

Program Comprehension for Domain-Specific Programming Languages

QUESTIONNAIRE 4

GraphViz

General-Purpose Language

Name:	
Class:	
University:	
Date:	

Start time:	
--------------------	--

Question 1

Marks: 0 / 1

QL011 GraphViz-GPL-UML: Please select correct GraphViz statements (without syntax errors):

Choose one answer.

a. `void createGraph1(Agraph_t *g) {`
`Agnode_t *person = agnode(g, "Person", "shape", "box");`
`agsafeset(person, "fontname", "Sans", "");`
`agsafeset(person, "fontsize", "8", "");`

`Agnode_t *account = agnode(g, "BankAccount", "shape", "box");`
`agsafeset(account, "fontname", "Sans", "");`
`agsafeset(account, "fontsize", "8", "");`

`Agedge_t person_account = agedge (g, person, account, "arrowhead", "open");`
`agsafeset(person_account, "headlabel", "1", "");`
`agsafeset(person_account, "fontname", "Sans", "");`
`agsafeset(person_account, "fontsize", "8", "");`
`}`

b. `void createGraph1(Agraph_t *g) {`
`Agnode_t *person = agnode(g, "Person");`
`agsafeset(person, "shape", "box", "");`
`agsafeset(person, "fontname", "Sans", "");`
`agsafeset(person, "fontsize", "8", "");`

`Agnode_t *account = agnode(g, "BankAccount");`
`agsafeset(account, "shape", "box", " ");`
`agsafeset(account, "fontname", "Sans", " ");`
`agsafeset(account, "fontsize", "8", " ");`

`Agedge_t *person_account = agedge (g, person, account);`
`agsafeset(person_account, "arrowhead", "open", "");`
`agsafeset(person_account, "headlabel", "1", "");`
`agsafeset(person_account, "fontname", "Sans", "");`
`agsafeset(person_account, "fontsize", "8", " ");`
`}`

c. `Agraph createGraph1(Agraph g) {`
`Agnode person := agnode(g, "Person");`
`agsafeset("person", "shape", "box", "");`
`agsafeset("person", "fontname", "Sans", "");`
`agsafeset("person", "fontsize", "8", "");`

`Agnode account := agnode(g, "BankAccount");`
`agsafeset("account", "shape", "box", " ");`
`agsafeset("account", "fontname", "Sans", " ");`
`agsafeset("account", "fontsize", "8", " ");`

`Agedge person_account := agedge (g, person, account);`
`agsafeset("person_account", "arrowhead", "open", "");`
`agsafeset("person_account", "headlabel", "1", " ");`
`agsafeset("person_account", "fontname", "Sans", " ");`
`agsafeset("person_account", "fontsize", "8", " ");`
`}`

```
d. void createGraph1(Agraph_t *g) {
    Agnode_t *person = agnode(g, "Person");
    agsafeset("person", "shape", "box");
    agsafeset("person", fontname, Sans);
    agsafeset("person", fontsize, 8);

    Agnode_t *account = agnode(g, "BankAccount");
    agsafeset("account", fontname, Sans);
    agsafeset("account", fontsize, 8);

    Agedge_t person_account = aedge (g, "person", "account");
    agsafeset("person_account", arrowhead, open);
    agsafeset("person_account", headlabel, 1);
    agsafeset("person_account", fontname, Sans);
    agsafeset("person_account", fontsize, 8);
}
```

```
e. Agraph createGraph1(Agraph g) {
    Agnode person = agnode(g, "Person");
    agset(person, "shape", "box");
    agset(person, "fontname", "Sans");
    agset(person, "fontsize", "8");

    Agnode account = agnode(g, "BankAccount");
    agset(account, "shape", "box");
    agset(account, "fontname", "Sans");
    agset(account, "fontsize", "8");

    Agedge person_account = aedge (g, person, account);
    agset(person_account, "arrowhead", "open");
    agset(person_account, "headlabel", "1");
    agset(person_account, "fontname", "Sans");
    agset(person_account, "fontsize", "8");
}
```

Question 2

Marks: 0 / 1

QL012 GraphViz-GPL-FlowChart: Please select correct GraphViz statements (without syntax errors):

Choose one answer.

```
a. void createGraph1a(Agraph_t *g) {
    Agnode_t *start = agnode(g, "Start");
    agsafeset(start, "shape", "box", "");
    agsafeset(start, "style", "rounded", "");
    agsafeset(start, "fixedsize", "true", "");
    agsafeset(start, "width", "2", "");
    agsafeset(start, "height", "0.4", "");

    Agnode_t *read = agnode(g, "Read a");
    agsafeset(read, "shape", "polygon", "");
    agsafeset(read, "skew", "0.7", "");
    agsafeset(read, "fixedsize", "true", "");
    agsafeset(read, "width", "2", "");
    agsafeset(read, "height", "0.4", "");

    Agnode_t *print = agnode(g, "Print a");
    agsafeset(print, "shape", "polygon", "");
    agsafeset(print, "skew", "0.7", "");
    agsafeset(print, "fixedsize", "true", "");
    agsafeset(print, "width", "2", "");
    agsafeset(print, "height", "0.4", "");

    Agnode_t *end = agnode(g, "End");
    agsafeset(end, "shape", "box", "");
    agsafeset(end, "style", "rounded", "");
}
```

```

agsafeset(end, "fixedsize", "true" , "");
agsafeset(end, "width" , "2" , "");
agsafeset(end, "height" , "0.4" , "");

Agedge_t *start_read = aedge (g, start, read);
agsafeset(start_read, "arrowhead", "open", "");

Agedge_t *read_print = aedge (g, read, print);
agsafeset(read_print, "arrowhead", "open", "");

Agedge_t *print_end = aedge (g, print, end);
agsafeset(print_end, "arrowhead", "open", "");
}

b. void createGraphlc(Agraph_t *g) {
Agnode_t *start = agnode(g, "Start" );
agsafeset(start, "shape" , "box" , "");
agsafeset(start, "style" , "rounded", "");
agsafeset(start, "fixedsize", "true" , "");
agsafeset(start, "width" , "2" , "");
agsafeset(start, "height" , "0.4" , "");

Agnode_t *read = agnode(g, "Read a" );
agsafeset(read, "shape" , "polygon", "");
agsafeset(read, "skew" , "0.7" , "");
agsafeset(read, "fixedsize", "true" , "");
agsafeset(read, "width" , "2" , "");
agsafeset(read, "height" , "0.4" , "");

Agnode_t *print = agnode(g, "Print a" );
agsafeset(print, "shape" , "polygon", "");
agsafeset(print, "skew" , "0.7" , "");
agsafeset(print, "fixedsize", "true" , "");
agsafeset(print, "width" , "2" , "");
agsafeset(print, "height" , "0.4" , "");

Agnode_t *end = agnode(g, "End" );
agsafeset(end, "shape" , "box" , "");
agsafeset(end, "style" , "rounded", "");
agsafeset(end, "fixedsize", "true" , "");
agsafeset(end, "width" , "2" , "");
agsafeset(end, "height" , "0.4" , "");

Agedge_t *start_read = aedge (g, start, read);
agsafeset(start_read, "arrowhead", "open", "");

Agedge_t *read_print = aedge (g, read, print);
agsafeset(read_print, "arrowhead", "open", "");

Agedge_t *print_end = aedge (g, print, end);
agsafeset(print_end, "arrowhead", "open", "");
}

c. void createGraphl(*g) {
start = agnode(g, "Start" );
agsafeset(start, "shape" , "box" , "");
agsafeset(start, "style" , "rounded", "");
agsafeset(start, "fixedsize", "true" , "");
agsafeset(start, "width" , "2" , "");
agsafeset(start, "height" , "0.4" , "");

read = agnode(g, "Read a" );
agsafeset(read, "shape" , "polygon", "");
agsafeset(read, "skew" , "0.7" , "");
agsafeset(read, "fixedsize", "true" , "");
agsafeset(read, "width" , "2" , "");
agsafeset(read, "height" , "0.4" , "");

```

```

start_read = aedge (g, start, read);
agsafeset(start_read, "arrowhead", "open", "");

print = agnode(g, "Print a" );
agsafeset(print, "shape" , "polygon", "");
agsafeset(print, "skew" , "0.7" , "");
agsafeset(print, "fixedsize", "true" , "");
agsafeset(print, "width" , "2" , "");
agsafeset(print, "height" , "0.4" , "");

read_print = aedge (g, read, print);
agsafeset(read_print, "arrowhead", "open", "");

end = agnode(g, "End" );
agsafeset(end, "shape" , "box" , "");
agsafeset(end, "style" , "rounded", "");
agsafeset(end, "fixedsize", "true" , "");
agsafeset(end, "width" , "2" , "");
agsafeset(end, "height" , "0.4" , "");

print_end = aedge (g, print, end);
agsafeset(print_end, "arrowhead", "open", "");
}

```

```

d. Agraph_t* createGraph1(*g) {
    start = agnode(g, "Start" );
    agsafeset(start, "shape" , "box" , "");
    agsafeset(start, "style" , "rounded", "");
    agsafeset(start, "fixedsize", "true" , "");
    agsafeset(start, "width" , "2" , "");
    agsafeset(start, "height" , "0.4" , "");

    read = agnode(g, "Read a" );
    agsafeset(read, "shape" , "polygon", "");
    agsafeset(read, "skew" , "0.7" , "");
    agsafeset(read, "fixedsize", "true" , "");
    agsafeset(read, "width" , "2" , "");
    agsafeset(read, "height" , "0.4" , "");

    *start_read = aedge (g, start, read);
    agsafeset(start_read, "arrowhead", "open", "");

```

```

    *print = agnode(g, "Print a" );
    agsafeset(print, "shape" , "polygon", "");
    agsafeset(print, "skew" , "0.7" , "");
    agsafeset(print, "fixedsize", "true" , "");
    agsafeset(print, "width" , "2" , "");
    agsafeset(print, "height" , "0.4" , "");

    *read_print = aedge (g, read, print);
    agsafeset(read_print, "arrowhead", "open", "");

    *end = agnode(g, "End" );
    agsafeset(end, "shape" , "box" , "");
    agsafeset(end, "style" , "rounded", "");
    agsafeset(end, "fixedsize", "true" , "");
    agsafeset(end, "width" , "2" , "");
    agsafeset(end, "height" , "0.4" , "");

    *print_end = aedge (g, print, end);
    agsafeset(print_end, "arrowhead", "open", "");
}

```

```

e. void createGraph1(Agraph_t *g) {
    Aedge_t *start = agnode(g, "Start" );
    agsafeset(start, "shape" , "box" , "");

```

```
    agsafeset(start, "style"      , "rounded", "");
    agsafeset(start, "fixedsize", "true"   , "");
    agsafeset(start, "width"     , "2"     , "");
    agsafeset(start, "height"    , "0.4"  , "");

    Agedge_t read = agnode(g, "Read a" );
    agsafeset(read, "shape"      , "polygon", "");
    agsafeset(read, "skew"       , "0.7"   , "");
    agsafeset(read, "fixedsize", "true"   , "");
    agsafeset(read, "width"     , "2"     , "");
    agsafeset(read, "height"    , "0.4"  , "");

    Agedge_t *edge = aedge (g, start, read);
    agsafeset(edge, "arrowhead", "open" , "");

    Agnode_t *print = agnode(g, "Print a" );
    agsafeset(print, "shape"     , "polygon", "");
    agsafeset(print, "skew"      , "0.7"   , "");
    agsafeset(print, "fixedsize", "true"   , "");
    agsafeset(print, "width"    , "2"     , "");
    agsafeset(print, "height"   , "0.4"  , "");

    Agedge_t *edge = aedge (g, read, print);
    agsafeset(edge, "arrowhead", "open" , "");

    Agnode_t *end = agnode(g, "End" );
    agsafeset(end, "shape"      , "box"   , "");
    agsafeset(end, "style"     , "rounded", "");
    agsafeset(end, "fixedsize", "true"   , "");
    agsafeset(end, "width"    , "2"     , "");
    agsafeset(end, "height"   , "0.4"  , "");

    Agedge_t *edge = aedge (g, print, end);
    agsafeset(edge, "arrowhead", "open" , "");
}
```

Question 3

Marks: 0 / 1

QL021 GraphViz-GPL-UML: Please select GraphViz program with no sense (unreasonable UML diagram):

Choose one answer.

- a. `void createGraph2(Agraph_t *g) {`
`Agnode_t *person = agnode(g, "{Person|- name : string\\l-`
`address : string\\l|+ getName() : string\\l+ getAddress() : string}");`
`agsafeset(person, "shape" , "record", "");`
`agsafeset(person, "fontname", "Sans" , "");`
`agsafeset(person, "fontsize", "8" , "");`
`}`
- b. `void createGraph2(Agraph_t *g) {`
`Agnode_t *person = agnode(g, "Person");`
`agsafeset(person, "shape" , "box" , "");`
`agsafeset(person, "fontname", "Sans", "");`
`agsafeset(person, "fontsize", "8" , "");`

`Agnode_t *address = agnode(g, "Address");`
`agsafeset(address, "shape" , "box" , "");`
`agsafeset(address, "fontname", "Sans", "");`
`agsafeset(address, "fontsize", "8" , "");`

`Agedge_t *person_address = agedge (g, person, address);`
`agsafeset(person_address, "arrowhead", "open", "");`
`agsafeset(person_address, "arrowtail", "none", "");`
`agsafeset(person_address, "headlabel", "1.n", "");`
`agsafeset(person_address, "fontname", "Sans", "");`
`agsafeset(person_address, "fontsize", "8" , "");`
`}`
- c. `void createGraph2(Agraph_t *g) {`
`Agnode_t *person = agnode(g, "Person");`
`agsafeset(person, "shape" , "box" , "");`
`agsafeset(person, "fontname", "Sans", "");`
`agsafeset(person, "fontsize", "8" , "");`

`Agnode_t *address = agnode(g, "Address");`
`agsafeset(address, "shape" , "box" , "");`
`agsafeset(address, "fontname", "Sans", "");`
`agsafeset(address, "fontsize", "8" , "");`

`Agedge_t *person_address = agedge (g, person, address);`
`agsafeset(person_address, "arrowhead", "open", "");`
`agsafeset(person_address, "arrowtail", "none", "");`
`agsafeset(person_address, "headlabel", "1" , "");`
`agsafeset(person_address, "fontname", "Sans", "");`
`agsafeset(person_address, "fontsize", "8" , "");`
`}`
- d. `void createGraph2(Agraph_t *g) {`
`Agnode_t *person = agnode(g, "Person");`
`agsafeset(person, "shape" , "box" , "");`
`agsafeset(person, "fontname", "Sans", "");`
`agsafeset(person, "fontsize", "8" , "");`

`Agnode_t *address = agnode(g, "Address");`
`agsafeset(address, "shape" , "box" , "");`
`agsafeset(address, "fontname", "Sans", "");`
`agsafeset(address, "fontsize", "8" , "");`

`Agedge_t *person_address = agedge (g, person, address);`
`agsafeset(person_address, "arrowhead", "none", "");`
`agsafeset(person_address, "arrowtail", "none", "");`
`}`

```

agsafeset(person_address, "headlabel", "1", "");
agsafeset(person_address, "taillabel", "1", "");
agsafeset(person_address, "fontname", "Sans", "");
agsafeset(person_address, "fontsize", "8", "");
}
e. void createGraph2(Agraph_t *g) {
    Agnode_t *person = agnode(g, "Person");
    agsafeset(person, "shape", "box", "");
    agsafeset(person, "fontname", "Sans", "");
    agsafeset(person, "fontsize", "8", "");

    Agnode_t *address = agnode(g, "Address");
    agsafeset(address, "shape", "box", "");
    agsafeset(address, "fontname", "Sans", "");
    agsafeset(address, "fontsize", "8", "");

    Agedge_t *person_address = agedge(g, person, address);
    agsafeset(person_address, "arrowtail", "empty", "");
    agsafeset(person_address, "arrowhead", "none", "");
    agsafeset(person_address, "fontname", "Sans", "");
    agsafeset(person_address, "fontsize", "8", "");
}

```

Question 4

Marks: 0 / 1

QL022 GraphViz-GPL-FlowChart: Please select GraphViz program with no sense (unreasonable Flowchart diagram):

Choose one answer.

```

a. void createGraph2(Agraph_t *g) {
    agnodeattr(g, "shape", "box");
    agnodeattr(g, "style", "rounded");
    agnodeattr(g, "fixedsize", "true");
    agnodeattr(g, "skew", "0.0");
    agnodeattr(g, "width", "2");
    agnodeattr(g, "height", "0.4");
    agedgeattr(g, "arrowhead", "open");

    Agnode_t *start = agnode(g, "Start");

    Agnode_t *print_text = agnode(g, "Print 'Write your name:'");
    agset(print_text, "shape", "polygon");
    agset(print_text, "style", "");
    agset(print_text, "skew", "0.7");
    agset(print_text, "width", "3");

    Agnode_t *read = agnode(g, "Read name");
    agset(read, "shape", "polygon");
    agset(read, "style", "");
    agset(read, "skew", "0.7");
    agset(read, "width", "3");

    Agnode_t *print_name = agnode(g, "Print 'Hello ' + name");
    agset(print_name, "shape", "polygon");
    agset(print_name, "style", "");
    agset(print_name, "skew", "0.7");
    agset(print_name, "width", "3");

    Agnode_t *end = agnode(g, "End");
    agset(end, "shape", "box");
    agset(end, "style", "rounded");
    agset(end, "width", "2");

    Agedge_t *start_printt = agedge(g, start, print_text);

```



```

Agedge_t *read_print = aedge (g, print_text, read );
Agedge_t *read_printn = aedge (g, read, print_name );
Agedge_t *printn_end = aedge (g, print_name, end );
}

b. void createGraph2(Agraph_t *g) {
    Agnode_t *end = agnode(g, "End" );
    agsafeset(end, "shape" , "box" , "");
    agsafeset(end, "style" , "rounded", "");
    agsafeset(end, "fixedsize", "true" , "");
    agsafeset(end, "fixedsize", "true" , "");
    agsafeset(end, "width" , "2" , "");
    agsafeset(end, "height" , "0.4" , "");

    Agnode_t *print_name = agnode(g, "Print 'Hello ' + name" );
    agsafeset(print_name, "shape" , "polygon", "");
    agsafeset(print_name, "skew" , "0.7" , "");
    agsafeset(print_name, "fixedsize", "true" , "");
    agsafeset(print_name, "width" , "3" , "");
    agsafeset(print_name, "height" , "0.4" , "");

    Agnode_t *read = agnode(g, "Read name" );
    agsafeset(read, "shape" , "polygon", "");
    agsafeset(read, "skew" , "0.7" , "");
    agsafeset(read, "fixedsize", "true" , "");
    agsafeset(read, "width" , "3" , "");
    agsafeset(read, "height" , "0.4" , "");

    Agnode_t *print_text = agnode(g, "Print 'Write your name:'" );
    agsafeset(print_text, "shape" , "polygon", "");
    agsafeset(print_text, "skew" , "0.7" , "");
    agsafeset(print_text, "fixedsize", "true" , "");
    agsafeset(print_text, "width" , "3" , "");
    agsafeset(print_text, "height" , "0.4" , "");

    Agnode_t *start = agnode(g, "Start" );
    agsafeset(start, "shape" , "box" , "");
    agsafeset(start, "style" , "rounded", "");
    agsafeset(start, "fixedsize", "true" , "");
    agsafeset(start, "width" , "2" , "");
    agsafeset(start, "height" , "0.4" , "");

    Agedge_t *printn_end = aedge (g, print_name, end);
    agsafeset(printn_end, "arrowhead", "open", "");

    Agedge_t *read_printn = aedge (g, read, print_name);
    agsafeset(read_printn, "arrowhead", "open", "");

    Agedge_t *read_print = aedge (g, print_text, read);
    agsafeset(read_print, "arrowhead", "open", "");

    Agedge_t *start_printt = aedge (g, start, print_text);
    agsafeset(start_printt, "arrowhead", "open", "");
}

c. void createGraph2(Agraph_t *g) {
    Agnode_t *start = agnode(g, "Start" );
    agsafeset(start, "shape" , "box" , "");
    agsafeset(start, "style" , "rounded", "");
    agsafeset(start, "fixedsize", "true" , "");
    agsafeset(start, "width" , "2" , "");
    agsafeset(start, "height" , "0.4" , "");

    Agnode_t *print_text = agnode(g, "Print 'Write your name:'" );
    agsafeset(print_text, "shape" , "polygon", "");
    agsafeset(print_text, "skew" , "0.7" , "");
    agsafeset(print_text, "fixedsize", "true" , "");

```

```

agsafeset(print_text, "width" , "3" , "");
agsafeset(print_text, "height" , "0.4" , "");

Agnode_t *read = agnode(g, "Read name" );
agsafeset(read, "shape" , "polygon", "");
agsafeset(read, "skew" , "0.7" , "");
agsafeset(read, "fixedsize", "true" , "");
agsafeset(read, "fixedsize", "true" , "");
agsafeset(read, "width" , "3" , "");
agsafeset(read, "height" , "0.4" , "");

Agnode_t *print_name = agnode(g, "Print 'Hello ' + name" );
agsafeset(print_name, "shape" , "polygon", "");
agsafeset(print_name, "skew" , "0.7" , "");
agsafeset(print_name, "fixedsize", "true" , "");
agsafeset(print_name, "width" , "3" , "");
agsafeset(print_name, "height" , "0.4" , "");

Agnode_t *end = agnode(g, "End" );
agsafeset(end, "shape" , "box" , "");
agsafeset(end, "style" , "rounded", "");
agsafeset(end, "fixedsize", "true" , "");
agsafeset(end, "width" , "2" , "");
agsafeset(end, "height" , "0.4" , "");

Agedge_t *start_printt = agedge (g, print_text, start);
agsafeset(start_printt, "arrowhead", "open", "");

Agedge_t *read_print = agedge (g, read, print_text);
agsafeset(read_print, "arrowhead", "open", "");

Agedge_t *read_printn = agedge (g, print_name, read);
agsafeset(read_printn, "arrowhead", "open", "");

Agedge_t *printn_end = agedge (g, end, print_name);
agsafeset(printn_end, "arrowhead", "open", "");
}
d. void createGraph2(Agraph_t *g) {
Agnode_t *end = agnode(g, "End" );
agsafeset(end, "shape" , "box" , "");
agsafeset(end, "style" , "rounded", "");
agsafeset(end, "fixedsize", "true" , "");
agsafeset(end, "fixedsize", "true" , "");
agsafeset(end, "width" , "2" , "");
agsafeset(end, "height" , "0.4" , "");

Agnode_t *print_name = agnode(g, "Print 'Hello ' + name" );
agsafeset(print_name, "shape" , "polygon", "");
agsafeset(print_name, "skew" , "0.7" , "");
agsafeset(print_name, "fixedsize", "true" , "");
agsafeset(print_name, "fixedsize", "true" , "");
agsafeset(print_name, "width" , "3" , "");
agsafeset(print_name, "height" , "0.4" , "");

Agnode_t *read = agnode(g, "Read name" );
agsafeset(read, "shape" , "polygon", "");
agsafeset(read, "skew" , "0.7" , "");
agsafeset(read, "fixedsize", "true" , "");
agsafeset(read, "fixedsize", "true" , "");
agsafeset(read, "width" , "3" , "");
agsafeset(read, "height" , "0.4" , "");

Agnode_t *print_text = agnode(g, "Print 'Write your name:'" );
agsafeset(print_text, "shape" , "polygon", "");
agsafeset(print_text, "skew" , "0.7" , "");

```

```

agsafeset(print_text, "fixedsize", "true" , "");
agsafeset(print_text, "fixedsize", "true" , "");
agsafeset(print_text, "width" , "3" , "");
agsafeset(print_text, "height" , "0.4" , "");

Agnode_t *start = agnode(g, "Start" );
agsafeset(start, "shape" , "box" , "");
agsafeset(start, "style" , "rounded", "");
agsafeset(start, "fixedsize", "true" , "");
agsafeset(start, "fixedsize", "true" , "");
agsafeset(start, "width" , "2" , "");
agsafeset(start, "height" , "0.4" , "");

Agedge_t *prinrn_end = agedge (g, print_name, end );
Agedge_t *read_prinrn = agedge (g, read, print_name );
Agedge_t *read_print = agedge (g, print_text, read );
Agedge_t *start_printt= agedge (g, start, print_text);
}

e. void createGraph2(Agraph_t *g) {
  agraphattr(g, "rankdir" , "TD" );
  agnodeattr(g, "shape" , "box" );
  agnodeattr(g, "style" , "rounded");
  agnodeattr(g, "fixedsize", "true" );
  agnodeattr(g, "skew" , "0.0" );
  agnodeattr(g, "width" , "2" );
  agnodeattr(g, "height" , "0.4" );
  agedgeattr(g, "arrowhead", "open" );

  Agnode_t *start = agnode(g, "Start");
  Agnode_t *print_text = agnode(g, "Print 'Write your name:'");
  Agnode_t *read = agnode(g, "Read name");
  Agnode_t *print_name = agnode(g, "Print 'Hello ' + name" );
  Agnode_t *end = agnode(g, "End");

  agsafeset(print_text, "shape" , "polygon" , "");
  agsafeset(print_text, "style" , " " , "");
  agsafeset(print_text, "skew" , "0.7" , "");
  agsafeset(print_text, "width" , "3" , "");
  agsafeset(read , "shape" , "polygon" , "");
  agsafeset(read , "style" , " " , "");
  agsafeset(read , "skew" , "0.7" , "");
  agsafeset(read , "width" , "3" , "");
  agsafeset(print_name, "shape" , "polygon" , "");
  agsafeset(print_name, "style" , " " , "");
  agsafeset(print_name, "skew" , "0.7" , "");
  agsafeset(print_name, "width" , "3" , "");
  agsafeset(end , "shape" , "box" , "");
  agsafeset(end , "style" , "rounded" , "");
  agsafeset(end , "width" , "2" , "");

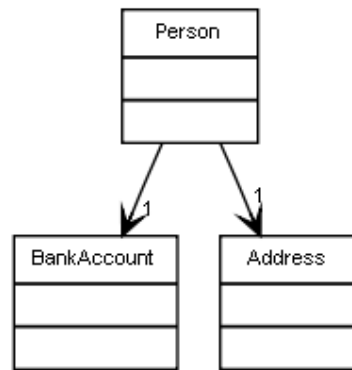
  Agedge_t *start_printt = agedge (g, start , print_text);
  Agedge_t *read_print = agedge (g, print_text, read );
  Agedge_t *read_prinrn = agedge (g, read , print_name);
  Agedge_t *prinrn_end = agedge (g, print_name, end );
}

```

Question 5

Marks: 0 / 1

QL031 GraphViz-GPL-UML: Select program for the following figure:



Choose one answer.

```

a. void createGraph3(Agraph_t *g) {
    Agnode_t *person = agnode(g, "{Person|\\1|\\1}");
    agsafeset(person, "shape", "record", "");
    agsafeset(person, "fontname", "Sans", "");
    agsafeset(person, "fontsize", "8", "");

    Agnode_t *account = agnode(g, "{BankAccount|\\1|\\1}");
    agsafeset(account, "shape", "record", "");
    agsafeset(account, "fontname", "Sans", "");
    agsafeset(account, "fontsize", "8", "");

    Agnode_t *address = agnode(g, "{Address|\\1|\\1}");
    agsafeset(address, "shape", "record", "");
    agsafeset(address, "fontname", "Sans", "");
    agsafeset(address, "fontsize", "8", "");
  }

```

```


    Agedge_t *person_account = agedge (g, person, account);
    agsafeset(person_account, "arrowhead", "open", "");
    agsafeset(person_account, "arrowtail", "none", "");
    agsafeset(person_account, "headlabel", "1..n", "");
    agsafeset(person_account, "labeldistance", "1.5", "");
    agsafeset(person_account, "fontname", "Sans", "");
    agsafeset(person_account, "fontsize", "8", "");

    Agedge_t *person_address = agedge (g, person, address);
    agsafeset(person_address, "arrowhead", "open", "");
    agsafeset(person_address, "arrowtail", "none", "");
    agsafeset(person_address, "headlabel", "1..n", "");
    agsafeset(person_address, "labeldistance", "1.5", "");
    agsafeset(person_address, "fontname", "Sans", "");
    agsafeset(person_address, "fontsize", "8", "");
  }

```

```

b. void createGraph3(Agraph_t *g) {
    Agnode_t *person = agnode(g, "{Person|\\1|\\1}");
    agsafeset(person, "shape", "record", "");
    agsafeset(person, "fontname", "Sans", "");
    agsafeset(person, "fontsize", "8", "");


    Agnode_t *account = agnode(g, "{BankAccount|\\1|\\1}");
    agsafeset(account, "shape", "record", "");
    agsafeset(account, "fontname", "Sans", "");
    agsafeset(account, "fontsize", "8", "");

    Agnode_t *address = agnode(g, "{Address|\\1|\\1}");

```

```

agsafeset(address, "shape" , "record", "");
agsafeset(address, "fontname", "Sans" , "");
agsafeset(address, "fontsize", "8" , "");

Agedge_t *address_person = agedge (g, address, person);
agsafeset(address_person, "arrowhead" , "open", "");
agsafeset(address_person, "arrowtail" , "none", "");
agsafeset(address_person, "headlabel" , "1" , "");
agsafeset(address_person, "labeldistance", "1.5" , "");
agsafeset(address_person, "fontname" , "Sans", "");
agsafeset(address_person, "fontsize" , "8" , "");

Agedge_t *address_account = agedge (g, address, account);
agsafeset(address_account, "arrowhead" , "open", "");
agsafeset(address_account, "arrowtail" , "none", "");
agsafeset(address_account, "headlabel" , "1" , "");
agsafeset(address_account, "labeldistance", "1.5" , "");
agsafeset(address_account, "fontname" , "Sans", "");
agsafeset(address_account, "fontsize" , "8" , "");
}

c. void createGraph3(Agraph_t *g) {
Agnode_t *person = agnode(g, "{Person|\\1|\\1}");
agsafeset(person, "shape" , "record", "");
agsafeset(person, "fontname", "Sans" , "");
agsafeset(person, "fontsize", "8" , "");

Agnode_t *account = agnode(g, "{BankAccount|\\1|\\1}");
agsafeset(account, "shape" , "record", "");
agsafeset(account, "fontname", "Sans" , "");
agsafeset(account, "fontsize", "8" , "");

Agnode_t *address = agnode(g, "{Address|\\1|\\1}");
agsafeset(address, "shape" , "record", "");
agsafeset(address, "fontname", "Sans" , "");
agsafeset(address, "fontsize", "8" , "");

Agedge_t *account_person = agedge (g, account, person);
agsafeset(account_person, "arrowhead" , "open", "");
agsafeset(account_person, "arrowtail" , "none", "");
agsafeset(account_person, "headlabel" , "1" , "");
agsafeset(account_person, "labeldistance", "1.5" , "");
agsafeset(account_person, "fontname" , "Sans", "");
agsafeset(account_person, "fontsize" , "8" , "");

Agedge_t *account_address = agedge (g, account, address);
agsafeset(account_address, "arrowhead" , "open", "");
agsafeset(account_address, "arrowtail" , "none", "");
agsafeset(account_address, "headlabel" , "1" , "");
agsafeset(account_address, "labeldistance", "1.5" , "");
agsafeset(account_address, "fontname" , "Sans", "");
agsafeset(account_address, "fontsize" , "8" , "");
}

d. void createGraph3(Agraph_t *g) {
Agnode_t *person = agnode(g, "{Person|\\1|\\1}");
agsafeset(person, "shape" , "record", "");
agsafeset(person, "fontname", "Sans" , "");
agsafeset(person, "fontsize", "8" , "");

Agnode_t *account = agnode(g, "{BankAccount|\\1|\\1}");
agsafeset(account, "shape" , "record", "");
agsafeset(account, "fontname", "Sans" , "");
agsafeset(account, "fontsize", "8" , "");

Agnode_t *address = agnode(g, "{Address|\\1|\\1}");
agsafeset(address, "shape" , "record", "");

```

```

agsafeset(address, "fontname", "Sans" , "");
agsafeset(address, "fontsize", "8" , "");

Agedge_t *person_account = agedge (g, person, account);
agsafeset(person_account, "arrowhead" , "open", "");
agsafeset(person_account, "arrowtail" , "none", "");
agsafeset(person_account, "headlabel" , "1" , "");
agsafeset(person_account, "labeldistance", "1.5" , "");
agsafeset(person_account, "fontname" , "Sans", "");
agsafeset(person_account, "fontsize" , "8" , "");

Agedge_t *person_address = agedge (g, person, address);
agsafeset(person_address, "arrowhead" , "open", "");
agsafeset(person_address, "arrowtail" , "none", "");
agsafeset(person_address, "headlabel" , "1" , "");
agsafeset(person_address, "labeldistance", "1.5" , "");
agsafeset(person_address, "fontname" , "Sans", "");
agsafeset(person_address, "fontsize" , "8" , "");
}

e. void createGraph3(Agraph_t *g) {
    Agnode_t *person = agnode(g, "Person" );
    agsafeset(person, "shape" , "box" , "");
    agsafeset(person, "fontname", "Sans", "");
    agsafeset(person, "fontsize", "8" , "");

    Agnode_t *account = agnode(g, "BankAccount");
    agsafeset(account, "shape" , "box" , "");
    agsafeset(account, "fontname", "Sans", "");
    agsafeset(account, "fontsize", "8" , "");

    Agnode_t *address = agnode(g, "Address" );
    agsafeset(address, "shape" , "box" , "");
    agsafeset(address, "fontname", "Sans", "");
    agsafeset(address, "fontsize", "8" , "");

    Agedge_t *account_person = agedge (g, account, person);
    agsafeset(account_person, "arrowhead" , "open", "");
    agsafeset(account_person, "arrowtail" , "none", "");
    agsafeset(account_person, "headlabel" , "1" , "");
    agsafeset(account_person, "labeldistance", "1.5" , "");
    agsafeset(account_person, "fontname" , "Sans", "");
    agsafeset(account_person, "fontsize" , "8" , "");

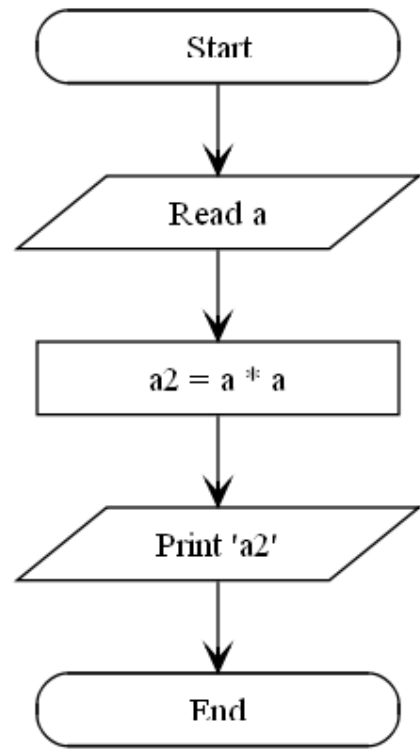
    Agedge_t *account_address = agedge (g, account, address);
    agsafeset(account_address, "arrowhead" , "open", "");
    agsafeset(account_address, "arrowtail" , "none", "");
    agsafeset(account_address, "headlabel" , "1" , "");
    agsafeset(account_address, "labeldistance", "1.5" , "");
    agsafeset(account_address, "fontname" , "Sans", "");
    agsafeset(account_address, "fontsize" , "8" , "");
}

```

Question 6

Marks: 0 / 1

QL032 GraphViz-GPL-FlowChart: Select program for the following figure:



Choose one answer.

- a. `void createGraph3(Agraph_t *g) {`
`Agnode_t *start = agnode(g, "Start");`
`agsafeset(start, "style", "rounded", "");`

`Agnode_t *read_a = agnode(g, "Read a");`
`agsafeset(read_a, "skew", "0.7", "");`

`Agnode_t *a_sq = agnode(g, "a2 = a * a");`
`agsafeset(a_sq, "skew", "0.0", "");`

`Agnode_t *print_sq = agnode(g, "Print 'a2'");`
`agsafeset(print_sq, "skew", "0.7", "");`

 `Agnode_t *end = agnode(g, "End");`
`agsafeset(end, "style", "rounded", "");`

`Agedge_t *start_reada = agedge (g, start, read_a);`
`agsafeset(start_reada, "arrowhead", "open", "");`

`Agedge_t *read_calc = agedge (g, read_a, a_sq);`
`agsafeset(read_calc, "arrowhead", "open", "");`

`Agedge_t *asq_print = agedge (g, a_sq, print_sq);`
`agsafeset(asq_print, "arrowhead", "open", "");`

`Agedge_t *printsq_end = agedge (g, print_sq, end);`
`agsafeset(printsq_end, "arrowhead", "open", "");`
`}`

```

b. void createGraph3(Agraph_t *g) {
    Agnode_t *start = agnode(g, "Start");
    agsafeset(start, "shape"      , "box"      , "");
    agsafeset(start, "style"      , "rounded" , "");
    agsafeset(start, "fixedsize"  , "true"    , "");
    agsafeset(start, "width"      , "2"       , "");
    agsafeset(start, "height"     , "0.4"    , "");

    Agnode_t *read_a = agnode(g, "Read a" );
    agsafeset(read_a, "shape"      , "polygon" , "");
    agsafeset(read_a, "skew"      , "0.7"    , "");
    agsafeset(read_a, "fixedsize" , "true"    , "");
    agsafeset(read_a, "width"     , "2"      , "");
    agsafeset(read_a, "height"    , "0.4"    , "");

    Agnode_t *a_sq = agnode(g, "a2 = a * a" );
    agsafeset(a_sq, "shape"       , "box"    , "");
    agsafeset(a_sq, "skew"       , "0.0"   , "");
    agsafeset(a_sq, "fixedsize"  , "true"   , "");
    agsafeset(a_sq, "width"      , "2"     , "");
    agsafeset(a_sq, "height"     , "0.4"   , "");

    Agnode_t *print_sq = agnode(g, "Print 'a2'");
    agsafeset(print_sq, "shape"   , "polygon", "");
    agsafeset(print_sq, "skew"   , "0.7"   , "");
    agsafeset(print_sq, "fixedsize", "true"   , "");
    agsafeset(print_sq, "width"  , "2"     , "");
    agsafeset(print_sq, "height" , "0.4"   , "");

    Agnode_t *end = agnode(g, "End" );
    agsafeset(end, "shape"       , "box"    , "");
    agsafeset(end, "style"       , "rounded", "");
    agsafeset(end, "fixedsize"  , "true"   , "");
    agsafeset(end, "width"      , "2"     , "");
    agsafeset(end, "height"     , "0.4"   , "");

    agedgeattr(g, "arrowhead", "open");
    Agedge_t *start_reada = agedge (g, start, read_a);
    Agedge_t *read_calc = agedge (g, read_a, a_sq);
    Agedge_t *asq_print = agedge (g, a_sq, print_sq);
    Agedge_t *printsq_end = agedge (g, print_sq, end);
}

```

```

c. void createGraph3(Agraph_t *g) {
    Agnode_t *start = agnode(g, "Start");
    agsafeset(start, "shape"      , "box"      , "");
    agsafeset(start, "style"      , "rounded" , "");
    agsafeset(start, "fixedsize"  , "true"    , "");
    agsafeset(start, "width"      , "2"       , "");
    agsafeset(start, "height"     , "0.4"    , "");

    Agnode_t *read_a = agnode(g, "Read a");
    agsafeset(read_a, "shape"      , "polygon" , "");
    agsafeset(read_a, "skew"      , "0.7"    , "");
    agsafeset(read_a, "fixedsize" , "true"    , "");
    agsafeset(read_a, "width"     , "2"      , "");
    agsafeset(read_a, "height"    , "0.4"    , "");

    Agnode_t *a_sq = agnode(g, "a2 = a * a");
    agsafeset(a_sq, "shape"       , "box"    , "");
    agsafeset(a_sq, "skew"       , "0.0"   , "");
    agsafeset(a_sq, "fixedsize"  , "true"   , "");
    agsafeset(a_sq, "width"      , "2"     , "");
    agsafeset(a_sq, "height"     , "0.4"   , "");
}

```



```

Agnode_t *print_sq = agnode(g, "Print 'a2'");
agsafeset(print_sq, "shape" , "polygon", "");
agsafeset(print_sq, "skew" , "0.7" , "");
agsafeset(print_sq, "fixedsize", "true" , "");
agsafeset(print_sq, "width" , "2" , "");
agsafeset(print_sq, "height" , "0.4" , "");

Agnode_t *end = agnode(g, "End" );
agsafeset(end, "shape" , "box" , "");
agsafeset(end, "style" , "rounded", "");
agsafeset(end, "fixedsize", "true" , "");
agsafeset(end, "width" , "2" , "");
agsafeset(end, "height" , "0.4" , "");

Agedge_t *start_asq = aedge (g, start, a_sq);
agsafeset(start_asq, "arrowhead", "open", "");

Agedge_t *asq_print = aedge (g, a_sq, print_sq);
agsafeset(asq_print, "arrowhead", "open", "");

Agedge_t *print_read = aedge (g, print_sq, read_a);
agsafeset(print_read, "arrowhead", "open", "");

Agedge_t *reada_end = aedge (g, read_a, end);
agsafeset(reada_end, "arrowhead", "open", "");
}
d. void createGraph3(Agraph_t *g) {
Agnode_t *start = agnode(g, "Start");
agsafeset(start, "shape" , "box" , "");
agsafeset(start, "style" , "rounded", "");

Agnode_t *read_a = agnode(g, "Read a");
agsafeset(read_a, "shape" , "polygon", "");
agsafeset(read_a, "skew" , "0.7" , "");

Agnode_t *a_sq = agnode(g, "a2 = a * a");
agsafeset(a_sq, "shape" , "box", "");
agsafeset(a_sq, "skew" , "0.0" , "");

Agnode_t *print_sq = agnode(g, "Print 'a2'");
agsafeset(print_sq, "shape" , "polygon", "");
agsafeset(print_sq, "skew" , "0.7" , "");

Agnode_t *end = agnode(g, "End" );
agsafeset(end, "shape" , "box" , "");
agsafeset(end, "style" , "rounded", "");

Agedge_t *start_reada = aedge (g, start, read_a);
agsafeset(start_reada, "arrowhead", "open", "");

Agedge_t *read_calc = aedge (g, read_a, a_sq);
agsafeset(read_calc, "arrowhead", "open", "");

Agedge_t *asq_print = aedge (g, a_sq, print_sq);
agsafeset(asq_print, "arrowhead", "open", "");

Agedge_t *printsq_end = aedge (g, print_sq, end);
agsafeset(printsq_end, "arrowhead", "open", "");
}

```

```
e. void createGraph3(Agraph_t *g) {
    Agnode_t *start = agnode(g, "Start");
    agsafeset(start, "shape"      , "box"      , "");
    agsafeset(start, "style"     , "rounded" , "");

    Agnode_t *read_b = agnode(g, "Read b");
    agsafeset(read_b, "shape"    , "polygon" , "");
    agsafeset(read_b, "skew"    , "0.7"    , "");

    Agnode_t *b_sq = agnode(g, "b2 = b * b");
    agsafeset(b_sq, "shape"     , "box"     , "");
    agsafeset(b_sq, "skew"     , "0.0"    , "");

    Agnode_t *print_sq = agnode(g, "Print 'b2'");
    agsafeset(print_sq, "shape"  , "polygon" , "");
    agsafeset(print_sq, "skew"  , "0.7"    , "");

    Agnode_t *end = agnode(g, "End");
    agsafeset(end, "shape"      , "box"      , "");
    agsafeset(end, "style"     , "rounded" , "");

    Agedge_t *start_reada = aedge (g, start, read_b);
    agsafeset(start_reada, "arrowhead", "open", "");

    Agedge_t *read_calc = aedge (g, read_b, b_sq);
    agsafeset(read_calc, "arrowhead", "open", "");

    Agedge_t *asq_print = aedge (g, b_sq, print_sq);
    agsafeset(asq_print, "arrowhead", "open", "");

    Agedge_t *printsq_end = aedge (g, print_sq, end);
    agsafeset(printsq_end, "arrowhead", "open", "");
}
```

End time:	
------------------	--

Question 7

Marks: 0 / 1

QC011 GraphViz-GPL-UML: Please select valid figure of the following GraphViz program:

```

void createGraph4(Agraph_t *g) {
    Agnode_t *family = agnode(g , "{Family|-
    surName : string\\|+ addPerson(Person aPerson) : void\\|+ currentState() : double\\|+
    getAddress(): string\\|+ getName() : string\\|+ income (integer aAmount) : double\\|
    l}");
    agsafeset(family, "shape"      , "record" , "");
    agsafeset(family, "fontname"   , "Sans"   , "");
    agsafeset(family, "fontsize"   , "8"      , "");

    Agnode_t *person = agnode(g , "{Person|-
    name : string\\|+ getName() : string\\|}");
    agsafeset(person, "shape"      , "record" , "");
    agsafeset(person, "fontname"   , "Sans"   , "");
    agsafeset(person, "fontsize"   , "8"      , "");

    Agnode_t *address = agnode(g , "{Address|- street: string\\|- city: string\\|-
    zipCode: integer\\|- country: string\\|+ getAddress(): string\\|}");
    agsafeset(address, "shape"      , "record" , "");
    agsafeset(address, "fontname"   , "Sans"   , "");
    agsafeset(address, "fontsize"   , "8"      , "");

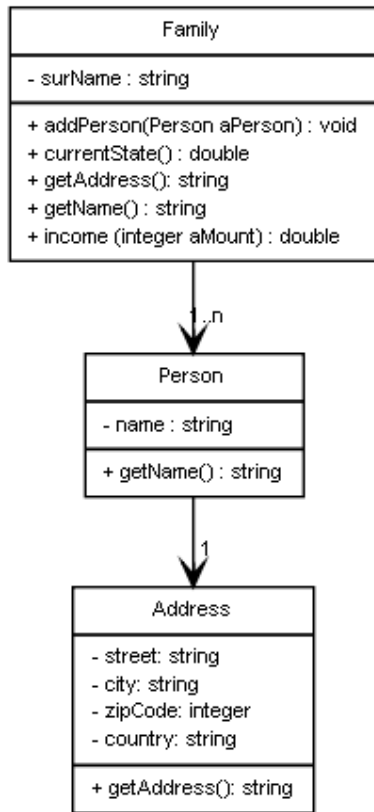
    Agedge_t *family_person = agedge (g, family, person );
    agsafeset(family_person, "arrowhead"      , "open" , "");
    agsafeset(family_person, "arrowtail"      , "odiamond" , "");
    agsafeset(family_person, "headlabel"      , "1..n" , "");
    agsafeset(family_person, "labeldistance"  , "1.5"  , "");
    agsafeset(family_person, "fontname"      , "Sans" , "");
    agsafeset(family_person, "fontsize"      , "8"    , "");

    Agedge_t *family_address = agedge (g, family, address);
    agsafeset(family_address, "arrowhead"      , "open" , "");
    agsafeset(family_address, "arrowtail"      , "odiamond" , "");
    agsafeset(family_address, "headlabel"      , "1"    , "");
    agsafeset(family_address, "labeldistance"  , "1.5"  , "");
    agsafeset(family_address, "fontname"      , "Sans" , "");
    agsafeset(family_address, "fontsize"      , "8"    , "");
}

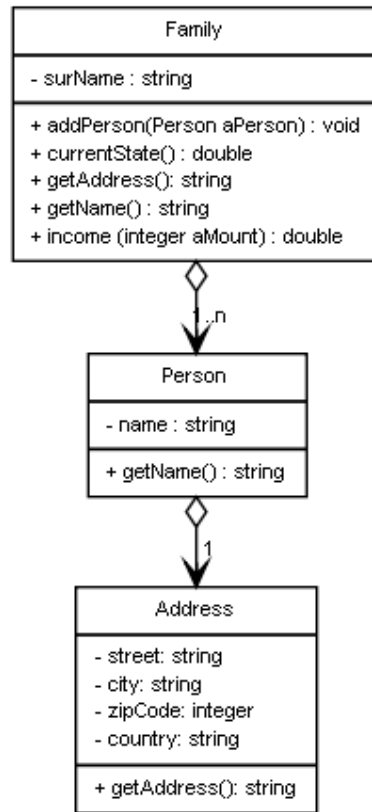
```

Choose one answer.

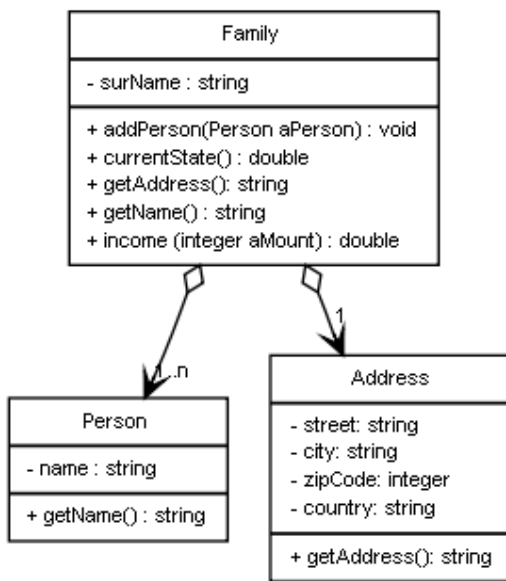
a.



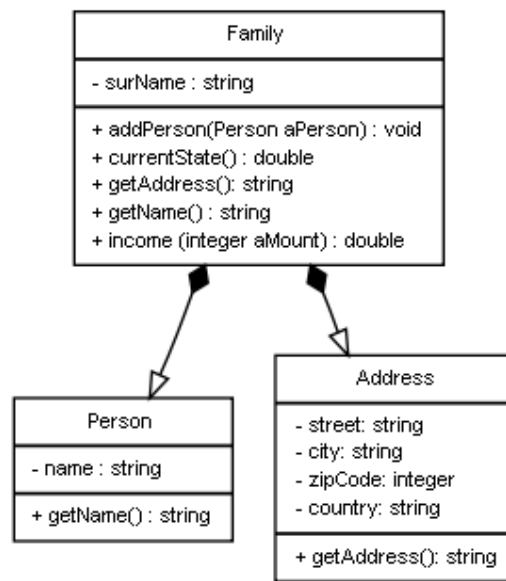
b.



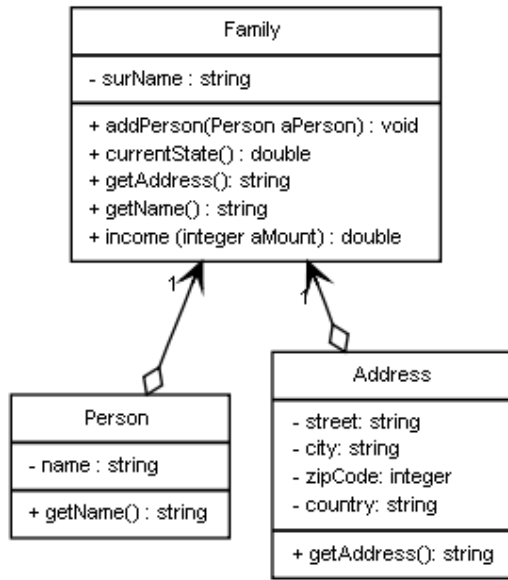
c.



d.



e.

**Question 8**

Marks: 0 / 1

QC012 GraphViz-GPL-FlowChart: Please select valid figure of the following GraphViz program:

```

void createGraph4(Agraph_t *g) {
    agnodeattr(g, "fixedsize", "true");
    agnodeattr(g, "width", "2");
    agnodeattr(g, "height", "0.4");

    aedgeattr(g, "arrowhead", "open");

    Agnode_t *start = agnode(g, "Start");
    agsafeset(start, "shape", "box", "");
    agsafeset(start, "style", "rounded", "");

    Agnode_t *print_text = agnode(g, "Print 'Input rectangle
                                length\n(l) and width (w)'" );
    agsafeset(print_text, "shape", "parallelogram", "");
    agsafeset(print_text, "width", "3", "");
    agsafeset(print_text, "height", "0.8", "");

    Agedge_t *start_reada = aedge (g, start, print_text);

    Agnode_t *read_l = agnode(g, "Read l");
    agsafeset(read_l, "shape", "parallelogram", "");

    Agedge_t *print_readl = aedge (g, print_text, read_l);

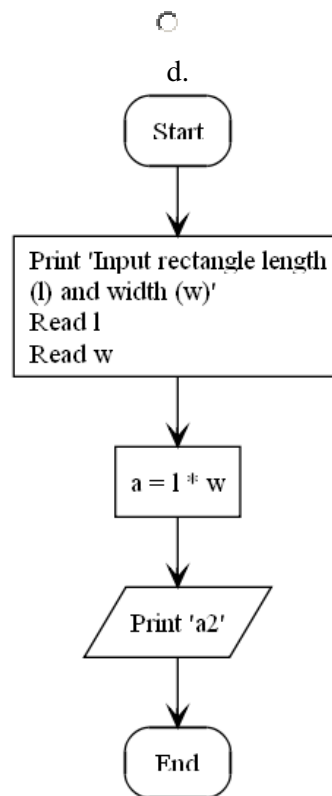
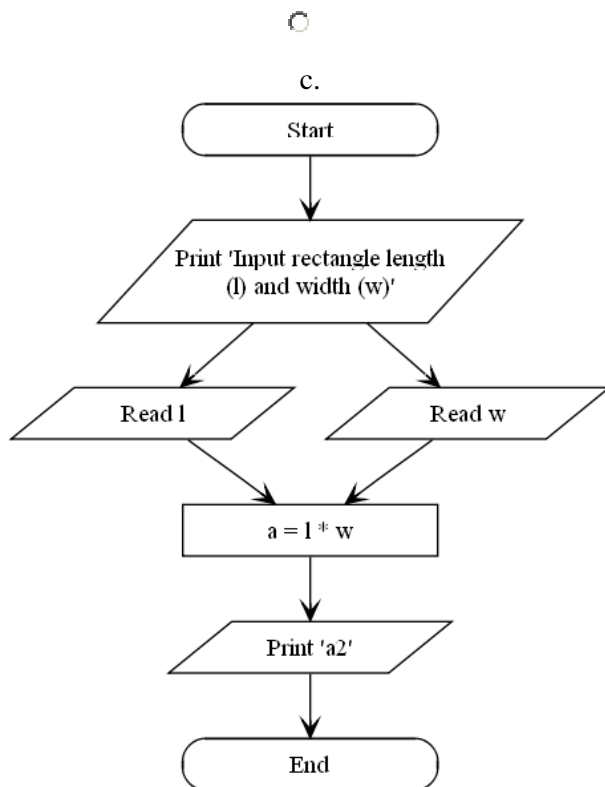
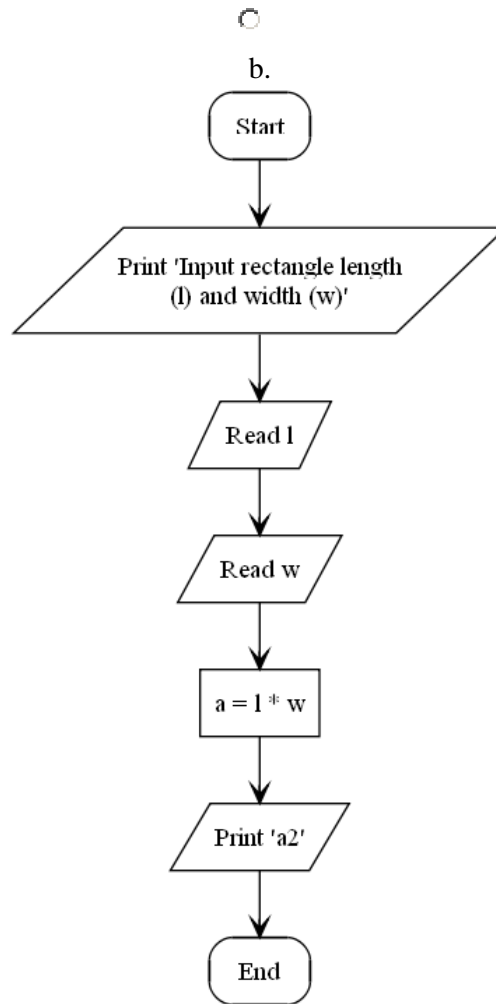
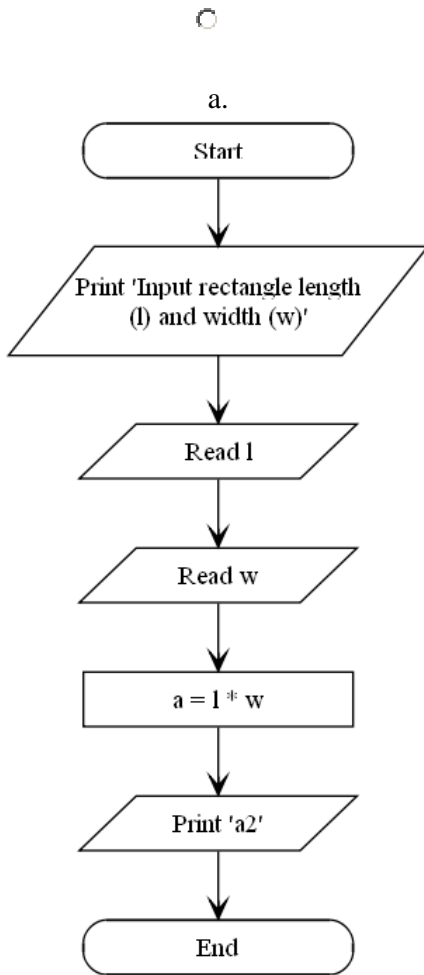
    Agnode_t *read_w = agnode(g, "Read w");
    agsafeset(read_w, "shape", "parallelogram", "");

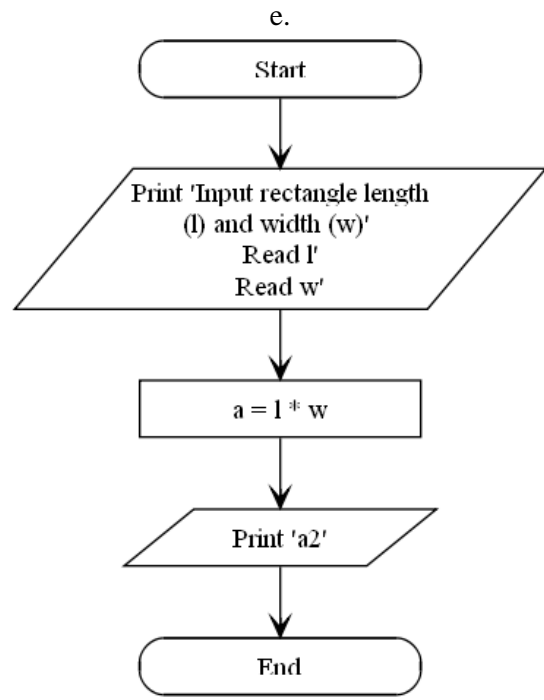
    Agedge_t *readl_readw = aedge (g, read_l, read_w);

    Agnode_t *calc = agnode(g, "a = l * w");
    agsafeset(calc, "shape", "box", "");

    Agedge_t *readw_calc = aedge (g, read_w, calc);
    Agnode_t *print_a = agnode(g, "Print 'a2'");
    agsafeset(print_a, "shape", "parallelogram", "");
    Agedge_t *calc_print = aedge (g, calc, print_a);
    Agnode_t *end = agnode(g, "End");
    agsafeset(end, "shape", "box", "");
    agsafeset(end, "style", "rounded", "");
    Agedge_t *print_end = aedge (g, print_a, end);
}
  
```

Choose one answer.



**Question 9**

Marks: 0 / 1

QC021 GraphViz-GPL-UML: Please select number of instance variables for class *BankAccount* :

```

void createGraph5(Agraph_t *g) {
    Agnode_t *account = agnode(g,
        "{BankAccount
        |
        - limit : double\\l
        - numberBA : integer\\l
        - person : Person\\l
        - transactions[] : double\\l
        - value: double\\l
        |
        + currentValue() : double\\l
        + getAccountOwner() : integer\\l
        + getNumberBA() : integer\\l
        + getLimit() : double\\l
        }");
    agsafeset(account, "shape", "record", "" );
    agsafeset(account, "fontname", "Sans", "" );
    agsafeset(account, "fontsize", "8", "" );
}
  
```

Choose one answer.

- a. 0
- b. 10
- c. 4
- d. 5
- e. 9

Question 10

Marks: 0 / 1

QC022 GraphViz-GPL-FlowChart: Please select number of nodes with *shape of diamond*:

```

void createGraph5(Agraph_t *g) {
    agnodeattr(g, "fixedsize", "true" );
    agnodeattr(g, "width" , "3" );
    agnodeattr(g, "height" , "0.4" );
    agnodeattr(g, "shape" , "diamond" );

    agedgeattr(g, "arrowhead", "open");

    Agnode_t *start = agnode(g, "Start");
    agsafeset(start, "shape", "box" , "");
    agsafeset(start, "style", "rounded", "");

    Agnode_t *print_text = agnode(g, "Print 'Input speed:');
    agsafeset(print_text, "shape", "parallelogram", "");

    Agnode_t *read_speed = agnode(g, "Read current_speed");
    agsafeset(read_speed, "shape", "parallelogram", "");

    Agnode_t *cond = agnode(g, "speed_limit >\ncurrent_speed");
    agsafeset(cond, "height", "1.0" , "");

    Agnode_t *print_thank = agnode(g, "Print 'Thank you!');
    agsafeset(print_thank, "shape", "parallelogram", "");

    Agnode_t *print_reduce = agnode(g, "Print 'Reduce speed!');
    agsafeset(print_reduce, "shape", "parallelogram", "");

    Agnode_t *end = agnode(g, "End");
    agsafeset(end, "shape", "box" , "");
    agsafeset(end, "style", "rounded" , "");

    Agedge_t *start_print = aedge (g, start , print_text);
    Agedge_t *print_readspeed = aedge (g, print_text, read_speed);
    Agedge_t *readspeed_cond = aedge (g, read_speed, cond );
    Agedge_t *cond_thank = aedge (g, cond , print_thank);
    agsafeset(cond_thank , "label", "Yes", "");
    Agedge_t *cond_reduce = aedge (g, cond , print_reduce);
    agsafeset(cond_reduce, "label", "No" , "");
    Agedge_t *thank_end = aedge (g, print_thank, end );
    Agedge_t *reduce_end = aedge (g, print_reduce, end );
}

```

Choose one answer.

- a. 4
 b. 0
 c. 3
 d. 1
 e. 2

Question 1 1

Marks: 0 / 1

Q031 GraphViz-GPL-UML: Select GraphViz programs with the same result as program below:

```

void createGraph6(Agraph_t *g) {
    Agnode_t *account = agnode(g, "{BankAccount|
                                   // attributes definition\\l|
                                   // methods definition\\l
                                   }");
    agsafeset(account, "shape"      , "record" , "");
    agsafeset(account, "fontname"   , "Sans"   , "");
    agsafeset(account, "fontsize"   , "8"      , "");

    Agnode_t *transaction = agnode(g, "{Transaction|
                                       // attributes definition\\l|
                                       // methods definition\\l
                                       }");
    agsafeset(transaction, "shape"    , "record" , "");
    agsafeset(transaction, "fontname" , "Sans"   , "");
    agsafeset(transaction, "fontsize" , "8"      , "");

    Agedge_t *account_transaction = agedge (g, account, transaction);
    agsafeset(account_transaction, "arrowhead"      , "open"      , "");
    agsafeset(account_transaction, "arrowtail"     , "odiamond"  , "");
    agsafeset(account_transaction, "headlabel"     , "0..n"     , "");
    agsafeset(account_transaction, "labeldistance" , "2.0"      , "");
    agsafeset(account_transaction, "minlen"       , "3.0"      , "");
    agsafeset(account_transaction, "fontname"     , "Sans"     , "");
    agsafeset(account_transaction, "fontsize"     , "8"        , "");
}

```

Choose one answer.

a.

```

void createGraph6(Agraph_t *g) {
    Agnode_t *account = agnode(g, "{BankAccount|
                                   // attributes definition\\l|
                                   // methods definition\\l
                                   }");
    agsafeset(account, "shape"      , "record" , "");
    agsafeset(account, "fontname"   , "Sans"   , "");
    agsafeset(account, "fontsize"   , "8"      , "");

    Agnode_t *account2 = agnode(g, "{BankAccount|
                                    // attributes definition\\l|
                                    // methods definition\\l
                                    }");
    agsafeset(account2, "shape"     , "record" , "");
    agsafeset(account2, "fontname"  , "Sans"   , "");
    agsafeset(account2, "fontsize"  , "8"      , "");

    Agnode_t *transaction = agnode(g, "{Transaction|
                                       // attributes definition\\l|
                                       // methods definition\\l
                                       }");
    agsafeset(transaction, "shape"    , "record" , "");
    agsafeset(transaction, "fontname" , "Sans"   , "");
    agsafeset(transaction, "fontsize" , "8"      , "");

    Agedge_t *account_transaction = agedge (g, account2, transaction);
    agsafeset(account_transaction, "arrowhead"      , "open"      , "");
    agsafeset(account_transaction, "arrowtail"     , "odiamond"  , "");
    agsafeset(account_transaction, "headlabel"     , "0..n"     , "");
    agsafeset(account_transaction, "labeldistance" , "2.0"      , "");
    agsafeset(account_transaction, "minlen"       , "3.0"      , "");
}

```

```

agsafeset(account_transaction, "fontname"      , "Sans"      , "");
agsafeset(account_transaction, "fontsize"     , "8"         , "");

Agedge_t *account_transaction2 = aedge (g, account, transaction);
agsafeset(account_transaction2, "arrowhead"    , "open"      , "");
agsafeset(account_transaction2, "arrowtail"   , "odiamond"  , "");
agsafeset(account_transaction2, "headlabel"   , "1..n"      , "");
agsafeset(account_transaction2, "labeldistance", "2.0"       , "");
agsafeset(account_transaction2, "minlen"     , "3.0"       , "");
agsafeset(account_transaction2, "fontname"    , "Sans"      , "");
agsafeset(account_transaction2, "fontsize"   , "8"         , "");
}

```

```

b. void createGraph6(Agraph_t *g) {
    Agnode_t *account = agnode(g, "{BankAccount|
        // attributes definition\\1|
        // methods definition\\1
    }");
    agsafeset(account, "shape"      , "record" , "");
    agsafeset(account, "fontname"   , "Sans"   , "");
    agsafeset(account, "fontsize"   , "8"      , "");

    Agnode_t *transaction = agnode(g, "{Transaction|
        // attributes definition\\1|
        // methods definition\\1
    }");
    agsafeset(transaction, "shape"    , "record" , "");
    agsafeset(transaction, "fontname" , "Sans"   , "");
    agsafeset(transaction, "fontsize" , "8"      , "");

    Agedge_t *transaction_account = aedge (g, transaction, account);
    agsafeset(transaction_account, "arrowhead"    , "open"      , "");
    agsafeset(transaction_account, "arrowtail"   , "odiamond"  , "");
    agsafeset(transaction_account, "headlabel"   , "0..n"      , "");
    agsafeset(transaction_account, "labeldistance", "2.0"       , "");
    agsafeset(transaction_account, "minlen"     , "3.0"       , "");
    agsafeset(transaction_account, "fontname"    , "Sans"      , "");
    agsafeset(transaction_account, "fontsize"   , "8"         , "");
}

```

```

c. void createGraph6(Agraph_t *g) {
    agsafeset(g, "shape"      , "record" , "");

    Agnode_t *account = agnode(g, "{BankAccount|
        // attributes definition\\1|
        // methods definition\\1
    }");
    agsafeset(account, "fontname" , "Sans"   , "");
    agsafeset(account, "fontsize" , "8"      , "");

    Agnode_t *transaction = agnode(g, "{Transaction|
        // attributes definition\\1|
        // methods definition\\1
    }");
    agsafeset(transaction, "fontname" , "Sans"   , "");
    agsafeset(transaction, "fontsize" , "8"      , "");

    Agedge_t *account_transaction2 = aedge (g, account, transaction);
    agsafeset(account_transaction2, "arrowhead"    , "open"      , "");
    agsafeset(account_transaction2, "arrowtail"   , "odiamond"  , "");
    agsafeset(account_transaction2, "headlabel"   , "1..n"      , "");
    agsafeset(account_transaction2, "labeldistance", "2.0"       , "");
    agsafeset(account_transaction2, "minlen"     , "3.0"       , "");
}

```

```

    agsafeset(account_transaction2, "fontname" , "Sans" , "");
    agsafeset(account_transaction2, "fontsize" , "8" , "");
}

```

```

d. void createGraph6(Agraph_t *g) {
    Agnode_t *account = agnode(g, "{BankAccount|
        // attributes definition\\1|
        // methods definition\\1
    }");
    agsafeset(account, "shape" , "record" , "");
    agsafeset(account, "fontname", "Sans" , "");
    agsafeset(account, "fontsize", "8" , "");

    Agnode_t *transaction = agnode(g, "{Transaction|
        // attributes definition\\1|
        // methods definition\\1
    }");
    agsafeset(transaction, "shape" , "record", "");
    agsafeset(transaction, "fontname", "Sans" , "");
    agsafeset(transaction, "fontsize", "8" , "");

    Agedge_t *account_transaction = agedge (g, account, transaction);
    agsafeset(account_transaction, "arrowhead" , "open" , "");
    agsafeset(account_transaction, "arrowtail" , "odiamond", "");
    agsafeset(account_transaction, "headlabel" , "0..n" , "");
    agsafeset(account_transaction, "labeldistance", "2.0" , "");
    agsafeset(account_transaction, "minlen" , "3.0" , "");
    agsafeset(account_transaction, "fontname" , "Sans" , "");
    agsafeset(account_transaction, "fontsize" , "8" , "");

    Agedge_t *account_transaction2 = agedge (g, account, transaction);
    agsafeset(account_transaction2, "arrowhead" , "open" , "");
    agsafeset(account_transaction2, "arrowtail" , "odiamond", "");
    agsafeset(account_transaction2, "headlabel" , "1..n" , "");
    agsafeset(account_transaction2, "labeldistance", "2.0" , "");
    agsafeset(account_transaction2, "minlen" , "3.0" , "");
    agsafeset(account_transaction2, "fontname" , "Sans" , "");
    agsafeset(account_transaction2, "fontsize" , "8" , "");
}

```

```

e. void createGraph6(Agraph_t *g) {
    Agnode_t *account = agnode(g, "{BankAccount|
        // attributes definition\\1|
        // methods definition\\1
    }");
    agsafeset(account, "shape" , "record" , "");
    agsafeset(account, "fontname", "Sans" , "");
    agsafeset(account, "fontsize", "8" , "");

    Agnode_t *account2 = agnode(g, "{BankAccount|
        // attributes definition\\1|
        // methods definition\\1
    }");
    agsafeset(account2, "shape" , "record" , "");
    agsafeset(account2, "fontname", "Sans" , "");
    agsafeset(account2, "fontsize", "8" , "");

    Agnode_t *transaction = agnode(g, "{Transaction|
        // attributes definition\\1|
        // methods definition\\1
    }");
    agsafeset(transaction, "shape" , "record", "");
    agsafeset(transaction, "fontname", "Sans" , "");
    agsafeset(transaction, "fontsize", "8" , "");
}

```

```
Agedge_t *account_transaction = agedge (g, account2, transaction);
agsafeset(account_transaction, "arrowhead"      , "open"      , "");
agsafeset(account_transaction, "arrowtail"     , "odiamond"  , "");
agsafeset(account_transaction, "headlabel"     , "0..n"      , "");
agsafeset(account_transaction, "labeldistance", "2.0"       , "");
agsafeset(account_transaction, "minlen"       , "3.0"       , "");
agsafeset(account_transaction, "fontname"     , "Sans"      , "");
agsafeset(account_transaction, "fontsize"     , "8"         , "");
}
```

Question 1 2

Marks: 0 / 1

Q032 GraphViz-GPL-FlowChart: Select GraphViz programs with the same result as program below:

```

void createGraph6(Agraph_t *g) {
    agraphattr(g, "rankdir" , "TD" );
    agraphattr(g, "ordering" , "out");

    agnodeattr(g, "fixedsize" , "true");
    agnodeattr(g, "width"      , "3" );
    agnodeattr(g, "height"     , "0.4" );
    agnodeattr(g, "shape"      , "parallelogram");

    agedgeattr(g, "arrowhead" , "open");

    Agnode_t *start = agnode(g, "Start");
    agsafeset(start, "shape" , "box" , "");
    agsafeset(start, "style" , "rounded", "");
    agsafeset(start, "width" , "2" , "");

    Agnode_t *print_text = agnode(g, "Print 'Input angle:");
    Agnode_t *read_angle = agnode(g, "Read angle");

    Agnode_t *cond1 = agnode(g, "a<90");
    agsafeset(cond1, "shape" , "diamond", "");
    agsafeset(cond1, "height" , "0.6" , "");
    agsafeset(cond1, "width" , "1.0" , "");

    Agnode_t *cond2 = agnode(g, "a<180");
    agsafeset(cond2, "shape" , "diamond", "");
    agsafeset(cond2, "height" , "0.6" , "");
    agsafeset(cond2, "width" , "1.0" , "");

    Agnode_t *print_acute = agnode(g, "Print 'Acute angle");
    Agnode_t *print_obtuse = agnode(g, "Print 'Obtuse angle");
    Agnode_t *print_reflex = agnode(g, "Print 'Reflex angle");

    Agnode_t *end = agnode(g, "End");
    agsafeset(end, "shape" , "box" , "");
    agsafeset(end, "style" , "rounded" , "");
    agsafeset(end, "width" , "2" , "");

    Agedge_t *start_print = agedge(g, start , print_text );
    Agedge_t *print_readangle = agedge(g, print_text, read_angle );
    Agedge_t *readangle_cond1 = agedge(g, read_angle, cond1 );
    Agedge_t *cond1_printacute = agedge(g, cond1 , print_acute );
    agsafeset(cond1_printacute, "label" , "Yes" , "");
    Agedge_t *cond1_cond2 = agedge(g, cond1 , cond2 );
    agsafeset(cond1_cond2, "label" , "No" , "");
    Agedge_t *cond2_printobtuse= agedge(g, cond2 , print_obtuse);
    agsafeset(cond2_printobtuse, "label" , "Yes" , "");
    Agedge_t *cond2_printreflex= agedge(g, cond2 , print_reflex);
    agsafeset(cond2_printreflex, "label" , "No" , "");
    Agedge_t *acute_end = agedge(g, print_acute , end );
    Agedge_t *obtuse_end = agedge(g, print_obtuse, end );
    Agedge_t *reflex_end = agedge(g, print_reflex, end );
}

```

Choose one answer.

- a. `void createGraph6(Agraph_t *g) {`
`agraphattr(g, "rankdir" , "TD");`
`agraphattr(g, "ordering" , "out");`

`agnodeattr(g, "fixedsize" , "true");`
`agnodeattr(g, "width" , "3");`

```

agnodeattr(g, "height" , "0.4" );
agnodeattr(g, "shape" , "parallelogram");

agedgeattr(g, "arrowhead", "open");

Agnode_t *start = agnode(g, "Start");
agsafeset(start, "shape", "box" , "");
agsafeset(start, "style", "rounded", "");
agsafeset(start, "width", "2" , "");

Agnode_t *print_text = agnode(g, "Print 'Input angle:'");
Agnode_t *read_angle = agnode(g, "Read angle");

Agnode_t *cond1 = agnode(g, "a<90");
agsafeset(cond1, "shape" , "diamond", "");
agsafeset(cond1, "height", "0.6" , "");
agsafeset(cond1, "width" , "1.0" , "");

Agnode_t *cond2 = agnode(g, "a<180");
agsafeset(cond2, "shape" , "diamond", "");
agsafeset(cond2, "height", "0.6" , "");
agsafeset(cond2, "width" , "1.0" , "");

Agnode_t *print_acute = agnode(g, "Print 'Acute angle'");
Agnode_t *print_obtuse = agnode(g, "Print 'Obtuse angle'");
Agnode_t *print_reflex = agnode(g, "Print 'Reflex angle'");

Agnode_t *end = agnode(g, "End");
agsafeset(end, "shape", "box" , "");
agsafeset(end, "style", "rounded" , "");
agsafeset(end, "width", "2" , "");

Agedge_t *start_print = aedge(g, start , print_text );
Agedge_t *print_readangle = aedge(g, print_text, read_angle );
Agedge_t *readangle_cond1 = aedge(g, read_angle, cond1 );
Agedge_t *cond1_printacute = aedge(g, cond1 , print_acute );
Agedge_t *cond1_cond2 = aedge(g, cond1 , cond2 );
Agedge_t *cond2_printobtuse = aedge(g, cond2 , print_obtuse);
Agedge_t *cond2_printreflex = aedge(g, cond2 , print_reflex);
Agedge_t *acute_end = aedge(g, print_acute , end );
Agedge_t *obtuse_end = aedge(g, print_obtuse, end );
Agedge_t *reflex_end = aedge(g, print_reflex, end );
}
b. void createGraph6(Agraph_t *g) {
  agraphattr(g, "rankdir" , "TD" );
  agraphattr(g, "ordering", "out");

  agnodeattr(g, "fixedsize", "true");
  agnodeattr(g, "width" , "3" );
  agnodeattr(g, "height" , "0.4" );

  Agnode_t *start = agnode(g, "Start");
  agsafeset(start, "shape", "box" , "");
  agsafeset(start, "style", "rounded", "");
  agsafeset(start, "width", "2" , "");

  Agnode_t *print_text = agnode(g, "Print 'Input angle:'");
  agsafeset(print_text, "shape", "parallelogram", "");

  Agnode_t *read_angle = agnode(g, "Read angle");
  agsafeset(read_angle, "shape", "parallelogram", "");

  Agnode_t *cond1 = agnode(g, "a<90");
  agsafeset(cond1, "shape" , "diamond", "");
  agsafeset(cond1, "height", "0.6" , "");

```

```

agsafeset(cond1, "width" , "1.0" , "");

Agnode_t *cond2 = agnode(g, "a<180");
agsafeset(cond2, "shape" , "diamond", "");
agsafeset(cond2, "height" , "0.6" , "");
agsafeset(cond2, "width" , "1.0" , "");

Agnode_t *print_acute = agnode(g, "Print 'Acute angle'");
agsafeset(print_acute , "shape", "parallelogram", "");

Agnode_t *print_obtuse = agnode(g, "Print 'Obtuse angle'");
agsafeset(print_obtuse, "shape", "parallelogram", "");

Agnode_t *print_reflex = agnode(g, "Print 'Reflex angle'");
agsafeset(print_reflex, "shape", "parallelogram", "");

Agnode_t *end = agnode(g, "End");
agsafeset(end, "shape", "box" , "");
agsafeset(end, "style", "rounded" , "");
agsafeset(end, "width", "2" , "");

Agedge_t *start_print = agedge(g, start , print_text );
agsafeset(start_print, "arrowhead", "open", "");
Agedge_t *print_readangle = agedge(g, print_text, read_angle );
agsafeset(print_readangle, "arrowhead", "open", "");
Agedge_t *readangle_cond1 = agedge(g, read_angle, cond1 );
agsafeset(readangle_cond1, "arrowhead", "open", "");
Agedge_t *cond1_printacute = agedge(g, cond1 , print_acute );
agsafeset(cond1_printacute, "label" , "Yes" , "");
agsafeset(cond1_printacute, "arrowhead", "open", "");
Agedge_t *cond1_cond2 = agedge(g, cond1 , cond2 );
agsafeset(cond1_cond2, "label" , "No" , "");
agsafeset(cond1_cond2, "arrowhead", "open", "");
Agedge_t *cond2_printobtuse= agedge(g, cond2 , print_obtuse);
agsafeset(cond2_printobtuse, "label" , "Yes" , "");
agsafeset(cond2_printobtuse, "arrowhead", "open", "");
Agedge_t *cond2_printreflex= agedge(g, cond2 , print_reflex);
agsafeset(cond2_printreflex, "label" , "No" , "");
agsafeset(cond2_printreflex, "arrowhead", "open", "");
Agedge_t *acute_end = agedge(g, print_acute , end );
agsafeset(acute_end, "arrowhead", "open", "");
Agedge_t *obtuse_end = agedge(g, print_obtuse, end );
agsafeset(obtuse_end, "arrowhead", "open", "");
Agedge_t *reflex_end = agedge(g, print_reflex, end );
agsafeset(reflex_end, "arrowhead", "open", "");
}

c. void createGraph6(Agraph_t *g) {
  agraphattr(g, "rankdir" , "TD" );
  agraphattr(g, "ordering", "out");

  agnodeattr(g, "fixedsize", "true");
  agnodeattr(g, "width" , "3" );
  agnodeattr(g, "height" , "0.4" );

  Agnode_t *start = agnode(g, "Start");
  agsafeset(start, "width", "2" , "");

  Agnode_t *print_text = agnode(g, "Print 'Input angle:');
  Agnode_t *read_angle = agnode(g, "Read angle");

  Agnode_t *cond1 = agnode(g, "a<90");
  agsafeset(cond1, "shape" , "diamond", "");
  agsafeset(cond1, "height" , "0.6" , "");
  agsafeset(cond1, "width" , "1.0" , "");

```

```

Agnode_t *cond2 = agnode(g, "a<180");
agsafeset(cond2, "shape", "diamond", "");
agsafeset(cond2, "height", "0.6", "");
agsafeset(cond2, "width", "1.0", "");

Agnode_t *print_acute = agnode(g, "Print 'Acute angle'");
Agnode_t *print_obtuse = agnode(g, "Print 'Obtuse angle'");
Agnode_t *print_reflex = agnode(g, "Print 'Reflex angle'");

Agnode_t *end = agnode(g, "End");
agsafeset(end, "width", "2", "");

Agedge_t *start_print = agedge(g, start, print_text);
agsafeset(start_print, "arrowhead", "open", "");
Agedge_t *print_readangle = agedge(g, print_text, read_angle);
agsafeset(print_readangle, "arrowhead", "open", "");
Agedge_t *readangle_cond1 = agedge(g, read_angle, cond1);
agsafeset(readangle_cond1, "arrowhead", "open", "");
Agedge_t *cond1_printacute = agedge(g, cond1, print_acute);
agsafeset(cond1_printacute, "label", "Yes", "");
agsafeset(cond1_printacute, "arrowhead", "open", "");
Agedge_t *cond1_cond2 = agedge(g, cond1, cond2);
agsafeset(cond1_cond2, "label", "No", "");
agsafeset(cond1_cond2, "arrowhead", "open", "");
Agedge_t *cond2_printobtuse = agedge(g, cond2, print_obtuse);
agsafeset(cond2_printobtuse, "label", "Yes", "");
agsafeset(cond2_printobtuse, "arrowhead", "open", "");
Agedge_t *cond2_printreflex = agedge(g, cond2, print_reflex);
agsafeset(cond2_printreflex, "label", "No", "");
agsafeset(cond2_printreflex, "arrowhead", "open", "");
Agedge_t *acute_end = agedge(g, print_acute, end);
agsafeset(acute_end, "arrowhead", "open", "");
Agedge_t *obtuse_end = agedge(g, print_obtuse, end);
agsafeset(obtuse_end, "arrowhead", "open", "");
Agedge_t *reflex_end = agedge(g, print_reflex, end);
agsafeset(reflex_end, "arrowhead", "open", "");
}

d. void createGraph6(Agraph_t *g) {
  agraphattr(g, "rankdir", "TD");
  agraphattr(g, "ordering", "out");

  agedgeattr(g, "arrowhead", "open");

  Agnode_t *start = agnode(g, "Start");
  agsafeset(start, "shape", "box", "");
  agsafeset(start, "style", "rounded", "");
  agsafeset(start, "width", "2", "");

  Agnode_t *print_text = agnode(g, "Print 'Input angle:'");
  agsafeset(print_text, "shape", "parallelogram", "");

  Agnode_t *read_angle = agnode(g, "Read angle");
  agsafeset(read_angle, "shape", "parallelogram", "");

  Agnode_t *cond1 = agnode(g, "a<90");
  agsafeset(cond1, "shape", "diamond", "");
  agsafeset(cond1, "height", "0.6", "");
  agsafeset(cond1, "width", "1.0", "");

  Agnode_t *cond2 = agnode(g, "a<180");
  agsafeset(cond2, "shape", "diamond", "");
  agsafeset(cond2, "height", "0.6", "");
  agsafeset(cond2, "width", "1.0", "");

  Agnode_t *print_acute = agnode(g, "Print 'Acute angle'");

```



```

agsafeset(print_acute , "shape", "parallelogram", "");

Agnode_t *print_obtuse = agnode(g, "Print 'Obtuse angle'");
agsafeset(print_obtuse, "shape", "parallelogram", "");

Agnode_t *end = agnode(g, "End");
agsafeset(end, "shape", "box", "");
agsafeset(end, "style", "rounded", "");
agsafeset(end, "width", "2", "");

Agedge_t *start_print = aedge(g, start , print_text );
Agedge_t *print_readangle = aedge(g, print_text, read_angle );
Agedge_t *readangle_cond1 = aedge(g, read_angle, cond1 );
Agedge_t *cond1_printacute = aedge(g, cond1 , print_acute );
agsafeset(cond1_printacute, "label", "Yes", "");
Agedge_t *cond1_cond2 = aedge(g, cond1 , cond2 );
agsafeset(cond1_cond2, "label", "No", "");
Agedge_t *cond2_printobtuse= aedge(g, cond2 , print_obtuse);
agsafeset(cond2_printobtuse, "label", "Yes", "");
Agedge_t *cond2_end = aedge(g, cond2 , end );
agsafeset(cond2_end, "label", "No", "");
Agedge_t *acute_end = aedge(g, print_acute , end );
Agedge_t *obtuse_end = aedge(g, print_obtuse, end );
}

e. void createGraph6(Agraph_t *g) {
    agnodeattr(g, "fixedsize", "true");
    agnodeattr(g, "width", "3");
    agnodeattr(g, "height", "0.4");
    agnodeattr(g, "shape", "parallelogram");

    aedgeattr(g, "arrowhead", "none");

    Agnode_t *start = agnode(g, "Start");
    agsafeset(start, "shape", "box", "");
    agsafeset(start, "style", "rounded", "");
    agsafeset(start, "width", "2", "");

    Agnode_t *print_text = agnode(g, "Print 'Input angle:');
    Agnode_t *read_angle = agnode(g, "Read angle");

    Agnode_t *cond1 = agnode(g, "a<90");
    agsafeset(cond1, "shape", "diamond", "");
    agsafeset(cond1, "height", "0.6", "");
    agsafeset(cond1, "width", "1.0", "");

    Agnode_t *cond2 = agnode(g, "a<180");
    agsafeset(cond2, "shape", "diamond", "");
    agsafeset(cond2, "height", "0.6", "");
    agsafeset(cond2, "width", "1.0", "");

    Agnode_t *print_acute = agnode(g, "Print 'Acute angle'");
    Agnode_t *print_obtuse = agnode(g, "Print 'Obtuse angle'");
    Agnode_t *print_reflex = agnode(g, "Print 'Reflex angle'");

    Agnode_t *end = agnode(g, "End");
    agsafeset(end, "shape", "box", "");
    agsafeset(end, "style", "rounded", "");
    agsafeset(end, "width", "2", "");

    Agedge_t *start_print = aedge(g, start , print_text );
    Agedge_t *print_readangle = aedge(g, print_text, read_angle );
    Agedge_t *readangle_cond1 = aedge(g, read_angle, cond1 );
    Agedge_t *cond1_printacute = aedge(g, cond1 , print_acute );
    agsafeset(cond1_printacute, "label", "Yes", "");
    Agedge_t *cond1_cond2 = aedge(g, cond1 , cond2 );

```

```
agsafeset(cond1_cond2, "label", "No", "");
Agedge_t *cond2_printobtuse= aedge(g, cond2      , print_obtuse);
agsafeset(cond2_printobtuse, "label", "Yes", "");
Agedge_t *cond2_printreflex= aedge(g, cond2      , print_reflex);
agsafeset(cond2_printreflex, "label", "No", "");
Agedge_t *acute_end      = aedge(g, print_acute , end      );
Agedge_t *obtuse_end     = aedge(g, print_obtuse, end     );
Agedge_t *reflex_end     = aedge(g, print_reflex, end     );
}
```

Question 13

Marks: 0 / 1

QC041 GraphViz-GPL-UML: Please select valid figure of the following GraphViz program. Notice, that the new construct *agfindattr* and *agxset* have been introduced to the program:

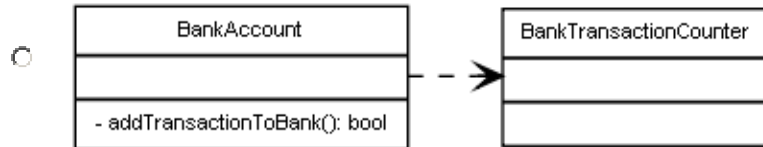
```
void createGraph7(Agraph_t *g) {
  agsafeset(g, "rankdir" , "LR" , "");
  Agnode_t *account = agnode(g, "BankAccount|
  \\|
  - addTransactionToBank(): bool\\|");

  agsafeset(account, "shape" , "record" , "");
  agsafeset(account, "fontname" , "Sans" , "");
  agsafeset(account, "fontsize" , "8" , "");
  Agnode_t *transaction_couter = agnode(g, "BankTransactionCounter|
  \\|
  \\|");

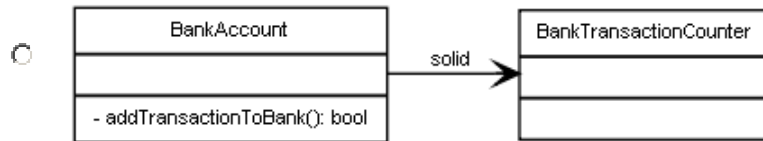
  agsafeset(transaction_couter, "shape" , "record" , "");
  agsafeset(transaction_couter, "fontname" , "Sans" , "");
  agsafeset(transaction_couter, "fontsize" , "8" , "");
  Agedge_t *account_counter = agedge (g, account, transaction_couter);
  agsafeset(account_counter, "arrowhead" , "open" , "");
  agsafeset(account_counter, "arrowtail" , "none" , "");
  agsafeset(account_counter, "style" , "solid" , "");
  agsafeset(account_counter, "fontname" , "Sans" , "");
  agsafeset(account_counter, "fontsize" , "8" , "");
  Agsym_t *a = agfindattr(account_counter, "style");
  agxset(account_counter, a->index, "dashed");
}
```

Choose one answer.

a.



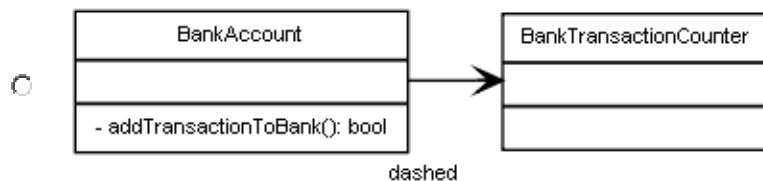
b.



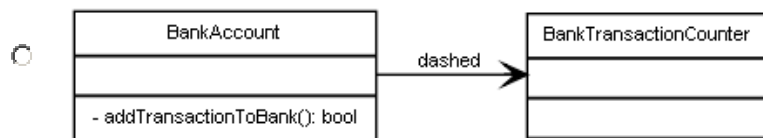
c.



d.



e.



Question 14

Marks: 0 / 1

QC042 GraphViz-GPL-FlowChart: Please select valid figure of the following GraphViz program. Notice, that the new construct *agget* has been introduced to the program:

```

void createGraph7(Agraph_t *g) {
    agraphattr(g, "rankdir" , "TD" );
    agraphattr(g, "ordering", "out");
    aedgeattr(g, "arrowhead", "open");

    Agnode_t *start = agnode(g, "Start");
    agsafeset(start, "shape", "box" , "");
    agsafeset(start, "style", "rounded", "");
    agsafeset(start, "width", "2" , "");

    Agnode_t *read_a = agnode(g, "Read a");
    agsafeset(read_a, "fixedsize", "true" , "");
    agsafeset(read_a, "width" , "2" , "");
    agsafeset(read_a, "height" , "0.4" , "");
    agsafeset(read_a, "shape" , "parallelogram", "");
    agsafeset(read_a, "peripheries", "2", "");

    Agnode_t *read_b = agnode(g, "Read b");
    agsafeset(read_b, "fixedsize", agget(read_a, "fixedsize"), "");
    agsafeset(read_b, "width" , agget(read_a, "width") , "");
    agsafeset(read_b, "height" , agget(read_a, "height") , "");
    agsafeset(read_b, "shape" , agget(read_a, "shape") , "");
    agsafeset(read_b, "peripheries", agget(read_a, "peripheries"), "");

    Agnode_t *cond1 = agnode(g, "a>b");
    agsafeset(cond1, "shape" , "diamond", "");
    agsafeset(cond1, "height", "0.6" , "");
    agsafeset(cond1, "width" , "1.0" , "");

    Agnode_t *print_a = agnode(g, "Print 'a'");
    agsafeset(print_a, "fixedsize", agget(read_a, "fixedsize"), "");
    agsafeset(print_a, "width" , agget(read_a, "width") , "");
    agsafeset(print_a, "height" , agget(read_a, "height") , "");
    agsafeset(print_a, "shape" , agget(read_a, "shape") , "");

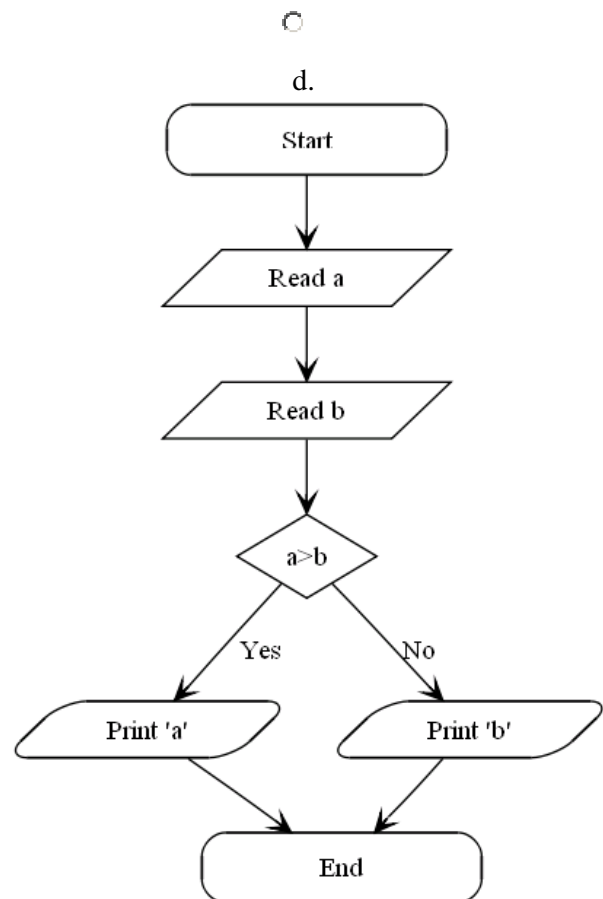
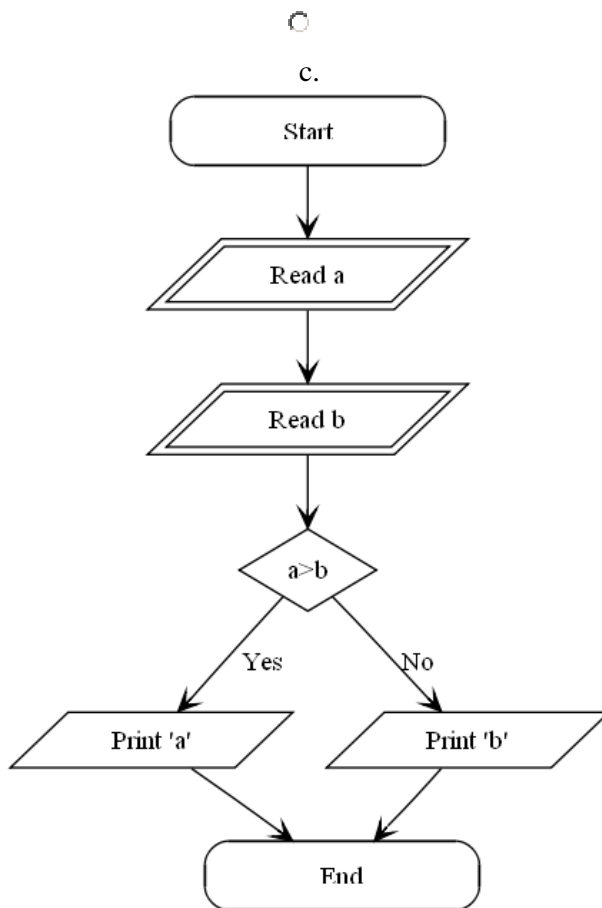
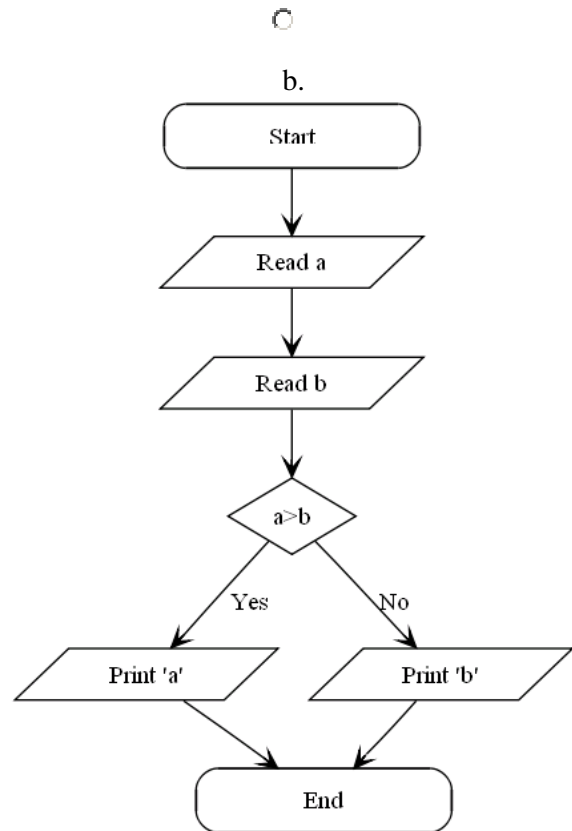
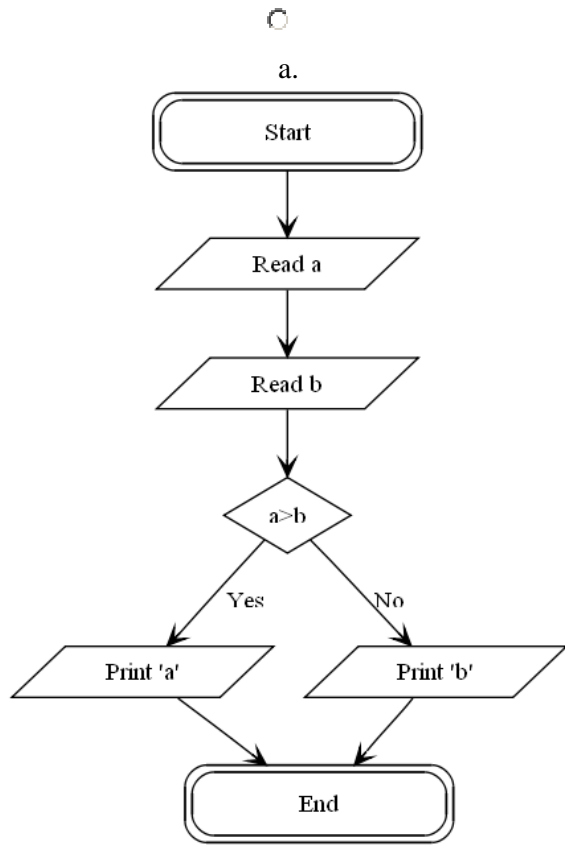
    Agnode_t *print_b = agnode(g, "Print 'b'");
    agsafeset(print_b, "fixedsize", agget(read_a, "fixedsize"), "");
    agsafeset(print_b, "width" , agget(read_a, "width") , "");
    agsafeset(print_b, "height" , agget(read_a, "height") , "");
    agsafeset(print_b, "shape" , agget(read_a, "shape") , "");

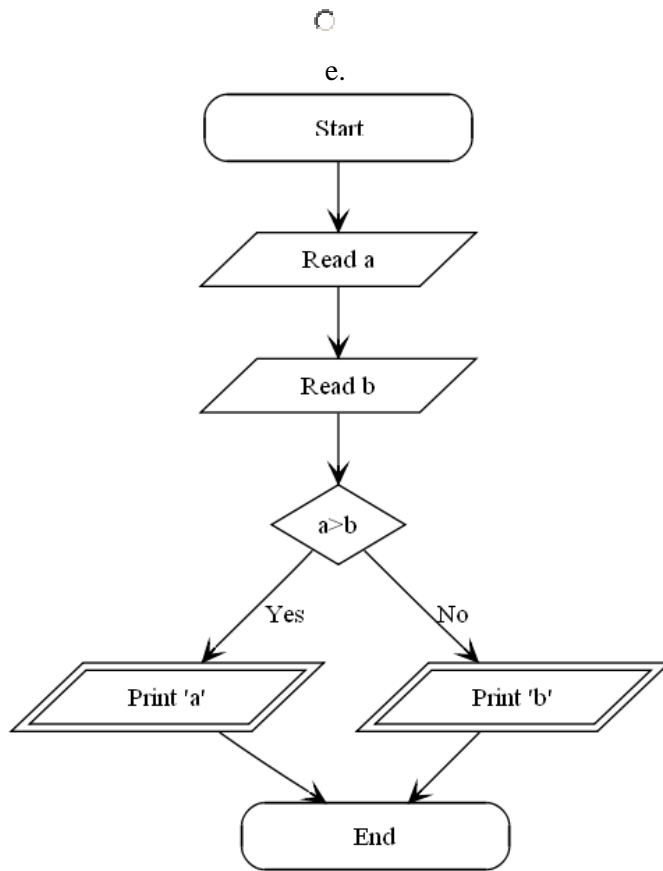
    Agnode_t *end = agnode(g, "End");
    agsafeset(end, "shape", agget(start, "shape") , "");
    agsafeset(end, "style", agget(start, "style") , "");
    agsafeset(end, "width", agget(start, "width") , "");

    Agedge_t *start_reada = aedge(g, start , read_a );
    Agedge_t *reada_readb = aedge(g, read_a, read_b );
    Agedge_t *readb_cond1 = aedge(g, read_b, cond1 );
    Agedge_t *cond1_cond2 = aedge(g, cond1 , print_a);
    agsafeset(cond1_cond2, "label", "Yes", "");
    Agedge_t *cond1_cond3 = aedge(g, cond1 , print_b);
    agsafeset(cond1_cond3, "label", "No", "");
    Agedge_t *printa_end = aedge(g, print_a, end );
    Agedge_t *printb_end = aedge(g, print_b, end );
}

```

Choose one answer.





Question 1 5

Marks: 0 / 1

QC051 GraphViz-GPL-UML: Please select number of classes inside package *Account.impl* in following GraphViz program:

```
void createGraph8(Agraph_t *g) {
    /*
    Definition of subgraph cluster_1. Note that Graphviz identifies the
    subgraph as a special "cluster" subgraph - nodes are contained within a
    bounding rectangle.
    */
    Agraph_t* g_sub = agsubg(g, "cluster_1");
    agsafeset(g_sub, "label" , "Package Account.impl" , "");
    agsafeset(g_sub, "fontname", "Sans", "");
    agsafeset(g_sub, "fontsize", "8" , "");

    /* Node "BankAccount" inside graph g */
    Agnode_t *account = agnode(g, "{BankAccount|\\1|\\1}");
    agsafeset(account, "shape" , "record", "");
    agsafeset(account, "fontname", "Sans" , "");
    agsafeset(account, "fontsize", "8" , "");

    /* Node "TransactionalAccount" inside subgraph cluster_1 */
    Agnode_t *t_account = agnode(g_sub, "{TransactionalAccount|\\1|\\1}");
    agsafeset(t_account, "shape" , "record", "");
    agsafeset(t_account, "fontname", "Sans" , "");
    agsafeset(t_account, "fontsize", "8" , "");

    /* Node "SavingAccount" inside subgraph cluster_1 */
    Agnode_t *saving_account = agnode(g_sub, "{SavingAccount|\\1|\\1}");
    agsafeset(saving_account, "shape" , "record", "");
    agsafeset(saving_account, "fontname", "Sans" , "");
    agsafeset(saving_account, "fontsize", "8" , "");

    /* Edge between "BankAccount" and "TransactionalAccount" */
    Agedge_t *ta_account = agedge (g, t_account, account );
    agsafeset(ta_account, "arrowhead", "empty", "");
    agsafeset(ta_account, "arrowtail", "none", "");
    agsafeset(ta_account, "minlen" , "3.0" , "");

    /* Edge between "BankAccount" and "SavingAccount" */
    Agedge_t *account_saving = agedge (g, account, saving_account);
    agsafeset(account_saving, "arrowhead", "none" , "");
    agsafeset(account_saving, "arrowtail", "empty", "");
    agsafeset(account_saving, "minlen" , "3.0" , "");
}
```

Choose one answer.

- a. 3
- b. 2
- c. 0
- d. 1
- e. 4

Question 16

Marks: 0 / 1

QC052 GraphViz-GPL-FlowChart: Please select number of nodes with *parallelogram* shape in the following program:

```

void createGraph8(Agraph_t *g) {
    /* All nodes have fixedsize */
    agnodeattr(g, "fixedsize", "true");
    /* Default width of all nodes is set to '2' */
    agnodeattr(g, "width", "2");
    /* Default height of all nodes is set to '0.4' */
    agnodeattr(g, "height", "0.4");
    /* Default node shape of the following
       program is 'parallelogram' */
    agnodeattr(g, "shape", "parallelogram");

    /* Default shape of edge head is open arrow */
    agedgeattr(g, "arrowhead", "open");

    /* Definition of node 'Start' */
    Agnode_t *start = agnode(g, "Start");
    agsafeset(start, "shape", "box", "");
    agsafeset(start, "style", "rounded", "");
    agsafeset(start, "width", "2", "");

    /* Definition of node 'Read n' */
    Agnode_t *read_n = agnode(g, "Read n");
    /* Definition of node 'sum = 0' */
    Agnode_t *init_sum = agnode(g, "sum = 0");

    /* Definition of node 'sum = sum + n' */
    Agnode_t *calc1 = agnode(g, "sum = sum + n");
    agsafeset(calc1, "shape", "box", "");
    agsafeset(calc1, "width", "2", "");

    /* Definition of node 'n = n - 1' */
    Agnode_t *calc2 = agnode(g, "n = n - 1");
    agsafeset(calc2, "shape", "box", "");
    agsafeset(calc2, "width", "2", "");

    /* Definition of node 'n>=0' */
    Agnode_t *cond1 = agnode(g, "n>=0");
    agsafeset(cond1, "shape", "diamond", "");
    agsafeset(cond1, "height", "0.6", "");
    agsafeset(cond1, "width", "1.0", "");

    /* Definition of node 'Print 'sum'' */
    Agnode_t *print_sum = agnode(g, "Print 'sum'");

    /* Definition of node 'End' */
    Agnode_t *end = agnode(g, "End");
    agsafeset(end, "shape", "box", "");
    agsafeset(end, "style", "rounded", "");

    /* Definition of edges between nodes */
    Agedge_t *start_readn = aedge(g, start, read_n);
    Agedge_t *readn_initsum = aedge(g, read_n, init_sum);
    Agedge_t *initsum_calc1 = aedge(g, init_sum, calc1);
    Agedge_t *calc1_calc2 = aedge(g, calc1, calc2);
    Agedge_t *calc2_cond1 = aedge(g, calc2, cond1);
    Agedge_t *cond1_printsum = aedge(g, cond1, print_sum);
    /* Definition of edge label between cond1 and print_sum */
    agsafeset(cond1_printsum, "label", "No", "");
    Agedge_t *cond1_calc1 = aedge(g, cond1, calc1);

```



```
/* Definition of edge label between cond1 and calc1 */
agsafeset(cond1_calc1, "label", "Yes", "");
/* Definition of edge head and tail ports set to east */
agsafeset(cond1_calc1, "tailport", "e", "");
agsafeset(cond1_calc1, "headport", "e", "");
Agedge_t *printsum_end = aedge(g, print_sum, end);
}
```

Choose one answer.

- a. 0
- b. 4
- c. 2
- d. 1
- e. 3

End time:	
------------------	--

Question 17

Marks: 0 / 1

QE011 GraphViz-GPL-UML: Expand class diagram with the connection between *Transaction* and *Withdraw* classes. Also, expand diagram with the connection between *Transaction* and *Deposit* classes. Both classes, *Withdraw* and *Deposit* are subclasses of class *Transaction* (use inheritance connection).

```

void createGraph9(Agraph_t *g) {
    Agraph_t* g_sub1 = agsubg(g, "");
    agsafeset(g_sub1, "rank", "same", "");

    Agraph_t* g_sub2 = agsubg(g, "cluster_2");
    agsafeset(g_sub2, "label", "Package Account.impl", "");
    agsafeset(g_sub2, "fontname", "Sans", "");
    agsafeset(g_sub2, "fontsize", "8", "");

    Agraph_t* g_sub3 = agsubg(g, "cluster_3");
    agsafeset(g_sub3, "label", "Package Transaction.impl", "");
    agsafeset(g_sub3, "fontname", "Sans", "");
    agsafeset(g_sub3, "fontsize", "8", "");

    Agnode_t *family = agnode(g, "{Family|
        - surName : string\\l|
        + addPerson(Person aPerson) : void\\l|
        + currentState() : double\\l|
        + getAddress(): string\\l|
        + getName() : string\\l|
        + income (integer aAmount) : double\\l|
    }");
    agsafeset(family, "shape", "record", "");
    agsafeset(family, "fontname", "Sans", "");
    agsafeset(family, "fontsize", "8", "");

    Agnode_t *address = agnode(g_sub1, "{Address|
        - street: string\\l|
        - city: string\\l|
        - zipCode: integer\\l|
        - country: string\\l|
        + getAddress(): string\\l|
    }");
    agsafeset(address, "shape", "record", "");
    agsafeset(address, "fontname", "Sans", "");
    agsafeset(address, "fontsize", "8", "");

    Agnode_t *person = agnode(g_sub1, "{Person|
        - name : string\\l|
        + getName() : string\\l|
    }");
    agsafeset(person, "shape", "record", "");
    agsafeset(person, "fontname", "Sans", "");
    agsafeset(person, "fontsize", "8", "");

    Agnode_t *account = agnode(g_sub1, "{BankAccount|
        - limit : double\\l|
        - numberBA : integer\\l|
        - value: double\\l|
        + currentValue() : double\\l|
        + getAccountOwner() : integer\\l|
        + getNumberBA() : integer\\l|
        + getLimit() : double\\l|
    }");
    agsafeset(account, "shape", "record", "");
    agsafeset(account, "fontname", "Sans", "");
    agsafeset(account, "fontsize", "8", "");

    Agnode_t *t_account = agnode(g_sub2, "{TransactionalAccount|\\l|\\l}");
    agsafeset(t_account, "shape", "record", "");

```

```

agsafeset(t_account, "fontname", "Sans" , "");
agsafeset(t_account, "fontsize", "8" , "");

Agnode_t *saving_account = agnode(g_sub2, "{SavingAccount|\\1|\\1}");
agsafeset(saving_account, "shape" , "record", "");
agsafeset(saving_account, "fontname", "Sans" , "");
agsafeset(saving_account, "fontsize", "8" , "");

Agnode_t *transaction = agnode(g_sub1, "{Transaction|
- value : integer\\1
- date : Date\\1|
+ getAccount() : BankAccount\\1
+ getDate() : Date\\1
+ getValue() : integer\\1
}");
agsafeset(transaction, "shape" , "record", "");
agsafeset(transaction, "fontname", "Sans" , "");
agsafeset(transaction, "fontsize", "8" , "");

Agnode_t *deposit = agnode(g_sub3, "{Deposit||}");
agsafeset(deposit, "shape" , "record", "");
agsafeset(deposit, "fontname", "Sans" , "");
agsafeset(deposit, "fontsize", "8" , "");

Agnode_t *withdraw = agnode(g_sub3, "{Withdraw||}");
agsafeset(withdraw, "shape" , "record", "");
agsafeset(withdraw, "fontname", "Sans" , "");
agsafeset(withdraw, "fontsize", "8" , "");

Agedge_t *family_address = agedge (g, family, address);
agsafeset(family_address, "arrowhead" , "open" , "");
agsafeset(family_address, "arrowtail" , "odiamond", "");
agsafeset(family_address, "headlabel" , "1" , "");
agsafeset(family_address, "labeldistance", "1.5" , "");
agsafeset(family_address, "fontname" , "Sans" , "");
agsafeset(family_address, "fontsize" , "8" , "");

Agedge_t *family_person = agedge (g, family, person);
agsafeset(family_person, "arrowhead" , "open" , "");
agsafeset(family_person, "arrowtail" , "odiamond", "");
agsafeset(family_person, "headlabel" , "1..n" , "");
agsafeset(family_person, "labeldistance", "1.5" , "");
agsafeset(family_person, "fontname" , "Sans" , "");
agsafeset(family_person, "fontsize" , "8" , "");

Agedge_t *person_account = agedge (g, person, account);
agsafeset(person_account, "arrowhead" , "open" , "");
agsafeset(person_account, "arrowtail" , "odiamond", "");
agsafeset(person_account, "headlabel" , "1" , "");
agsafeset(person_account, "labeldistance", "1.5" , "");
agsafeset(person_account, "fontname" , "Sans" , "");
agsafeset(person_account, "fontsize" , "8" , "");
agsafeset(person_account, "minlen" , "3.0" , "");

Agedge_t *account_transaction = agedge (g, account, transaction);
agsafeset(account_transaction, "arrowhead" , "none" , "");
agsafeset(account_transaction, "arrowtail" , "none" , "");
agsafeset(account_transaction, "headlabel" , "0..n" , "");
agsafeset(account_transaction, "taillabel" , "1" , "");
agsafeset(account_transaction, "labeldistance", "2.0" , "");
agsafeset(account_transaction, "minlen" , "3.0" , "");
agsafeset(account_transaction, "fontname" , "Sans" , "");
agsafeset(account_transaction, "fontsize" , "8" , "");

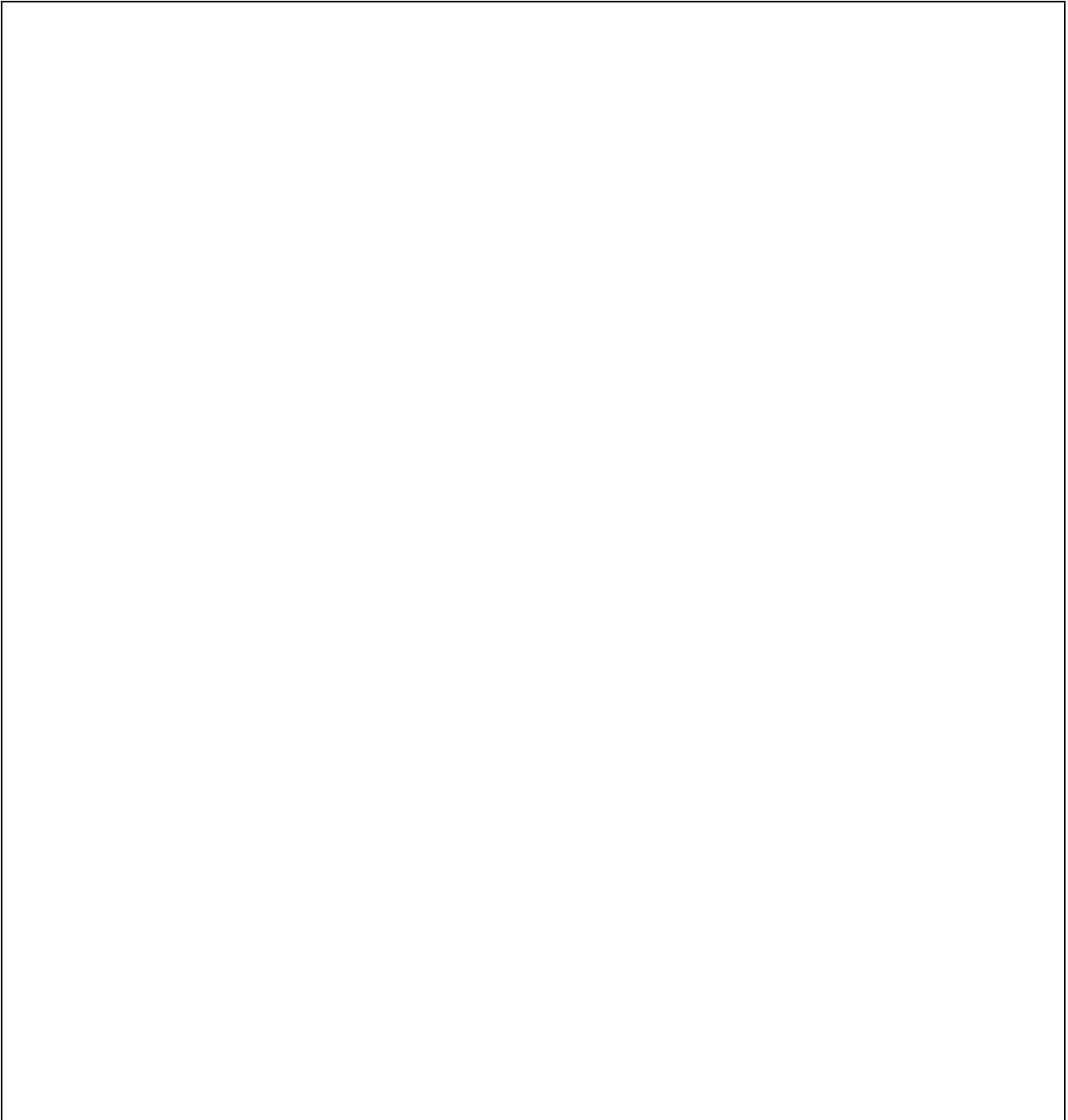
Agedge_t *ta_account = agedge (g, t_account, account);
agsafeset(ta_account, "arrowhead" , "empty", "");

```

```
agsafeset(ta_account, "arrowtail" , "none" , "");

Agedge_t *account_saving = aedge (g, account, saving_account);
agsafeset(account_saving, "arrowhead", "none" , "");
agsafeset(account_saving, "arrowtail", "empty", "");
}
```

Answer:



Question 18

Marks: 0 / 1

QE012 GraphViz-GPL-FlowChart: Expand flowchart diagram with the connection between *condition* $a \geq 0$ and *statement* $a = a - 1$. Also, supply the connection with label "Yes".

```

void createGraph9(Agraph_t *g) {
    agnodeattr(g, "fixedsize", "true");
    agnodeattr(g, "width", "2");
    agnodeattr(g, "height", "0.4");

    agedgeattr(g, "arrowhead", "open");

    Agnode_t *start = agnode(g, "Start");
    agsafeset(start, "shape", "box", "");
    agsafeset(start, "style", "rounded", "");
    agsafeset(start, "width", "2", "");

    Agnode_t *read_a = agnode(g, "Read a");
    agsafeset(read_a, "shape", "parallelogram", "");

    Agnode_t *calc1 = agnode(g, "a = a - 1");
    agsafeset(calc1, "shape", "box", "");
    agsafeset(calc1, "width", "2", "");

    Agnode_t *print_a = agnode(g, "Print 'a'");
    agsafeset(print_a, "shape", "parallelogram", "");

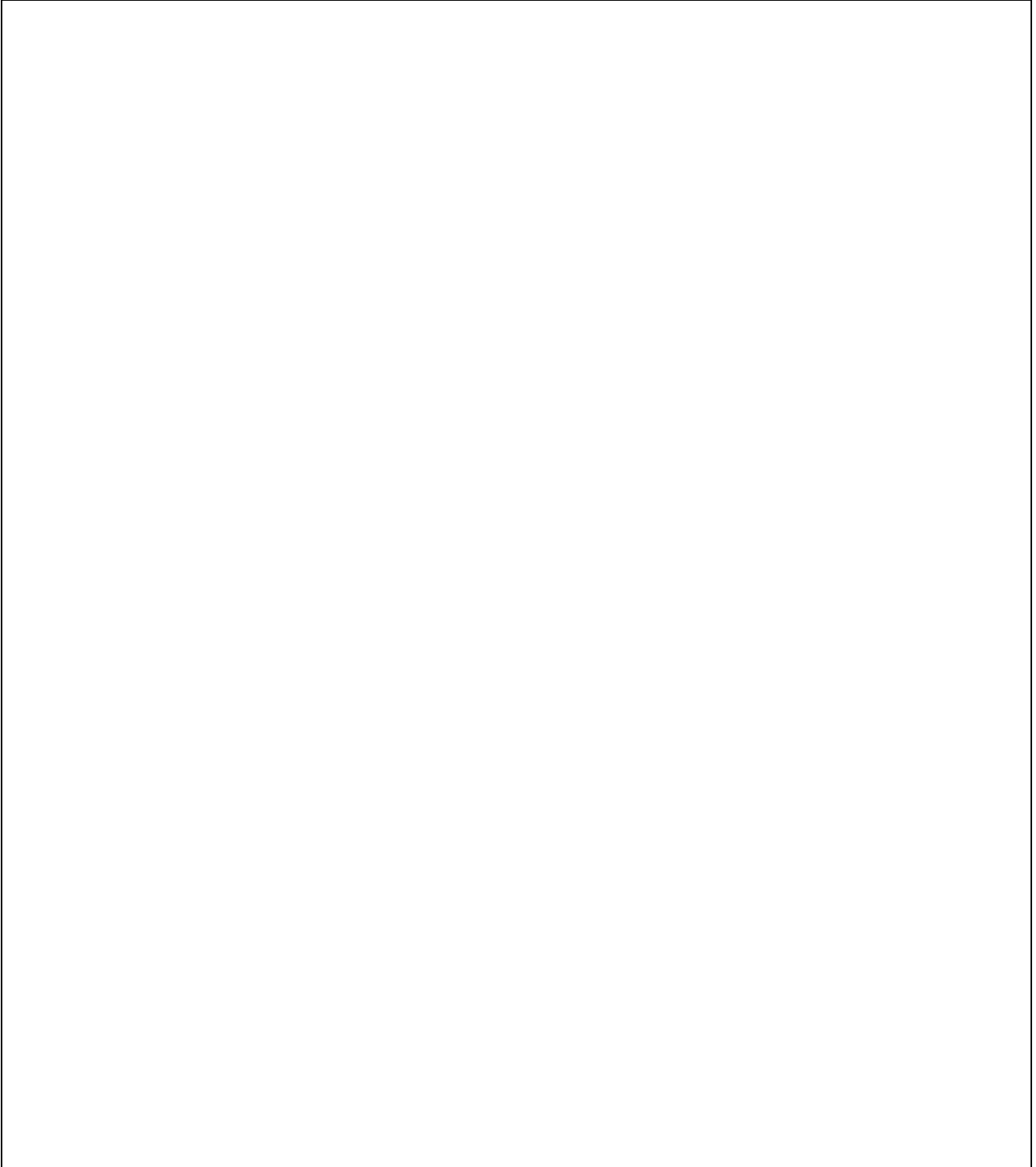
    Agnode_t *cond_1 = agnode(g, "a >= 0");
    agsafeset(cond_1, "shape", "diamond", "");
    agsafeset(cond_1, "height", "0.6", "");
    agsafeset(cond_1, "width", "1.0", "");

    Agnode_t *end = agnode(g, "End");
    agsafeset(end, "shape", "box", "");
    agsafeset(end, "style", "rounded", "");

    Agedge_t *start_reada = agedge(g, start, read_a);
    Agedge_t *reada_calc1 = agedge(g, read_a, calc1);
    Agedge_t *calc1_printa = agedge(g, calc1, print_a);
    Agedge_t *printa_cond_1 = agedge(g, print_a, cond_1);
    Agedge_t *cond1_end = agedge(g, cond_1, end);
    agsafeset(cond1_end, "label", "No", "");
}

```

Answer:

A large, empty rectangular box with a thin black border, intended for the user to provide an answer to the question above it.

Question 19

Marks: 0 / 1

QE021 GraphViz-GPL-UML: Change the GraphViz program below in a way classes *Family* and *Address* are no longer part of UML diagram. Also remove/change all edges connected with classes *Family* and *Address*.

```

void createGraph10(Agraph_t *g) {
    Agraph_t* g_sub1 = agsubg(g, "");
    agsafeset(g_sub1, "rank", "same", "");

    Agraph_t* g_sub2 = agsubg(g, "cluster_2");
    agsafeset(g_sub2, "label", "Package Account.impl", "");
    agsafeset(g_sub2, "fontname", "Sans", "");
    agsafeset(g_sub2, "fontsize", "8", "");

    Agnode_t *family = agnode(g, "{Family|
        - surName : string\\l|
        + addPerson(Person aPerson) : void\\l|
        + currentState() : double\\l|
        + getAddress(): string\\l|
        + getName() : string\\l|
        + income (integer aMount) : double\\l|
        }");
    agsafeset(family, "shape", "record", "");
    agsafeset(family, "fontname", "Sans", "");
    agsafeset(family, "fontsize", "8", "");

    Agnode_t *address = agnode(g_sub1, "{Address|
        - street: string\\l|
        - city: string\\l|
        - zipCode: integer\\l|
        - country: string\\l|
        + getAddress(): string\\l|
        }");
    agsafeset(address, "shape", "record", "");
    agsafeset(address, "fontname", "Sans", "");
    agsafeset(address, "fontsize", "8", "");

    Agnode_t *person = agnode(g_sub1, "{Person|
        - name : string\\l|
        + getName() : string\\l|
        }");
    agsafeset(person, "shape", "record", "");
    agsafeset(person, "fontname", "Sans", "");
    agsafeset(person, "fontsize", "8", "");

    Agnode_t *account = agnode(g_sub1, "{BankAccount|
        - limit : double\\l|
        - numberBA : integer\\l|
        - value: double\\l|
        + currentValue() : double\\l|
        + getAccountOwner() : integer\\l|
        + getNumberBA() : integer\\l|
        + getLimit() : double\\l|
        }");
    agsafeset(account, "shape", "record", "");
    agsafeset(account, "fontname", "Sans", "");
    agsafeset(account, "fontsize", "8", "");

    Agnode_t *t_account = agnode(g_sub2, "{TransactionalAccount|\\l|\\l}");
    agsafeset(t_account, "shape", "record", "");
    agsafeset(t_account, "fontname", "Sans", "");
    agsafeset(t_account, "fontsize", "8", "");

    Agnode_t *saving_account = agnode(g_sub2, "{SavingAccount|\\l|\\l}");
    agsafeset(saving_account, "shape", "record", "");

```

```

agsafeset(saving_account, "fontname", "Sans" , "");
agsafeset(saving_account, "fontsize", "8" , "");

Agnode_t *transaction = agnode(g_sub1, "{Transaction|
    - value : integer\\l
    - date : Date\\l|
    + getAccount() : BankAccount\\l
    + getDate() : Date\\l
    + getValue() : integer\\l
}");
agsafeset(transaction, "shape" , "record", "");
agsafeset(transaction, "fontname", "Sans" , "");
agsafeset(transaction, "fontsize", "8" , "");

Agedge_t *family_address = agedge (g, family, address);
agsafeset(family_address, "arrowhead" , "open" , "");
agsafeset(family_address, "arrowtail" , "odiamond", "");
agsafeset(family_address, "headlabel" , "1" , "");
agsafeset(family_address, "labeldistance", "1.5" , "");
agsafeset(family_address, "fontname" , "Sans" , "");
agsafeset(family_address, "fontsize" , "8" , "");

Agedge_t *family_person = agedge (g, family, person);
agsafeset(family_person, "arrowhead" , "open" , "");
agsafeset(family_person, "arrowtail" , "odiamond", "");
agsafeset(family_person, "headlabel" , "1..n" , "");
agsafeset(family_person, "labeldistance", "1.5" , "");
agsafeset(family_person, "fontname" , "Sans" , "");
agsafeset(family_person, "fontsize" , "8" , "");

Agedge_t *person_account = agedge (g, person, account);
agsafeset(person_account, "arrowhead" , "open" , "");
agsafeset(person_account, "arrowtail" , "odiamond", "");
agsafeset(person_account, "headlabel" , "1" , "");
agsafeset(person_account, "labeldistance", "1.5" , "");
agsafeset(person_account, "fontname" , "Sans" , "");
agsafeset(person_account, "fontsize" , "8" , "");
agsafeset(person_account, "minlen" , "3.0" , "");

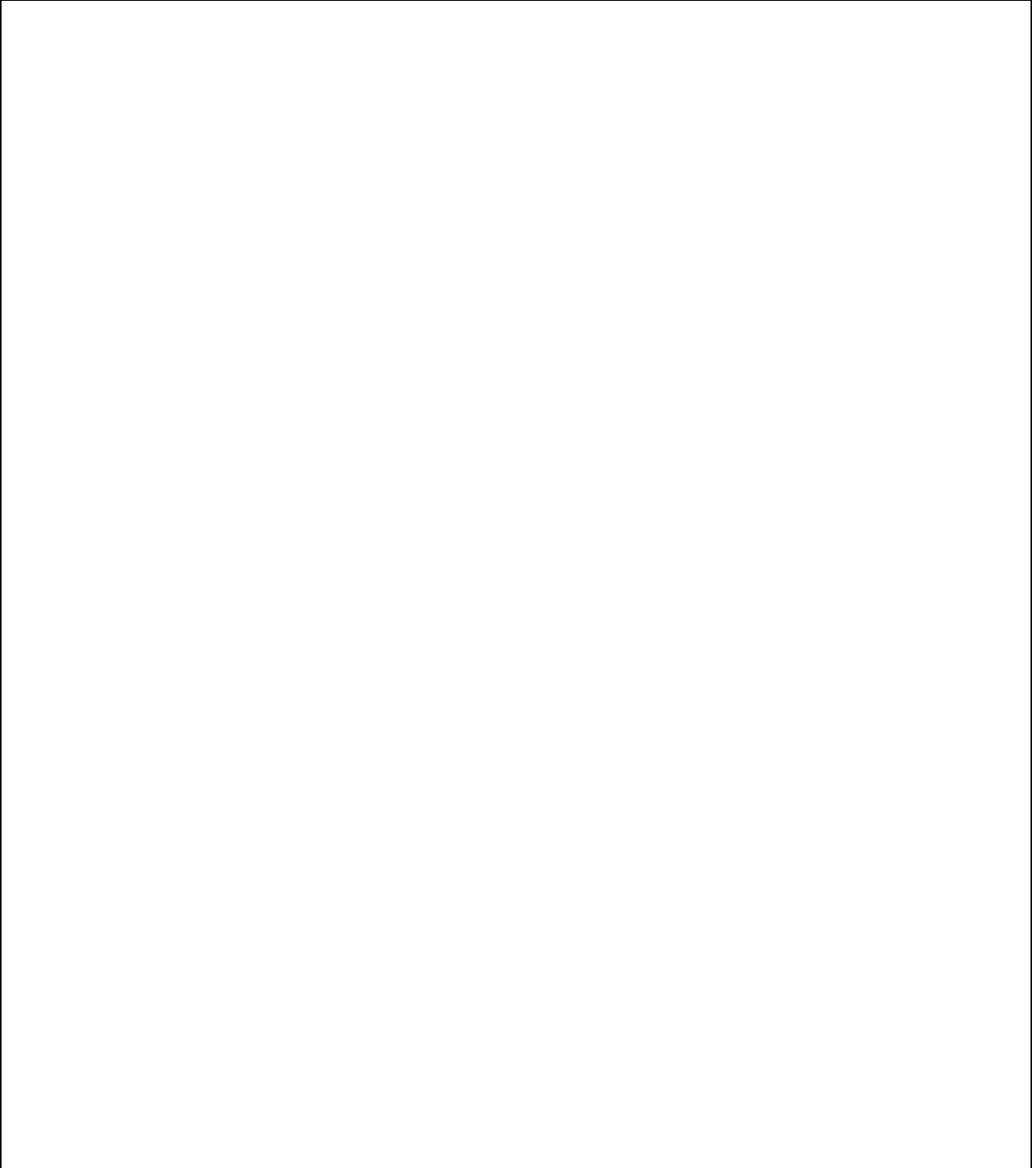
Agedge_t *account_transaction = agedge (g, account, transaction);
agsafeset(account_transaction, "arrowhead" , "none", "");
agsafeset(account_transaction, "arrowtail" , "none", "");
agsafeset(account_transaction, "headlabel" , "0..n" , "");
agsafeset(account_transaction, "taillabel" , "1" , "");
agsafeset(account_transaction, "labeldistance", "2.0" , "");
agsafeset(account_transaction, "minlen" , "3.0" , "");
agsafeset(account_transaction, "fontname" , "Sans" , "");
agsafeset(account_transaction, "fontsize" , "8" , "");

Agedge_t *ta_account = agedge (g, t_account, account);
agsafeset(ta_account, "arrowhead" , "empty", "");
agsafeset(ta_account, "arrowtail" , "none" , "");

Agedge_t *account_saving = agedge (g, account, saving_account);
agsafeset(account_saving, "arrowhead", "none" , "");
agsafeset(account_saving, "arrowtail", "empty", "");
}

```


Answer:



Question 20

Marks: 0 / 1

QE022 GraphViz-GPL-FlowChart: Change the GraphViz program below in a way that statement *Print 'n'* is no longer part of flowchart diagram. Also remove/change all edges connected with this node.

```

void createGraph10(Agraph_t *g) {
    agnodeattr(g, "fixedsize", "true");
    agnodeattr(g, "width", "2");
    agnodeattr(g, "height", "0.4");
    agnodeattr(g, "shape", "parallelogram");

    agedgeattr(g, "arrowhead", "open");

    Agnode_t *start = agnode(g, "Start");
    agsafeset(start, "shape", "box", "");
    agsafeset(start, "style", "rounded", "");
    agsafeset(start, "width", "2", "");

    Agnode_t *read_n = agnode(g, "Read n");
    Agnode_t *print_n= agnode(g, "Print 'n'");
    Agnode_t *init_f = agnode(g, "f = 1");

    Agnode_t *calc1 = agnode(g, "f = f * n");
    agsafeset(calc1, "shape", "box", "");
    agsafeset(calc1, "width", "2", "");

    Agnode_t *calc2 = agnode(g, "n = n - 1");
    agsafeset(calc2, "shape", "box", "");
    agsafeset(calc2, "width", "2", "");

    Agnode_t *cond1 = agnode(g, "n>0");
    agsafeset(cond1, "shape", "diamond", "");
    agsafeset(cond1, "height", "0.6", "");
    agsafeset(cond1, "width", "1.0", "");

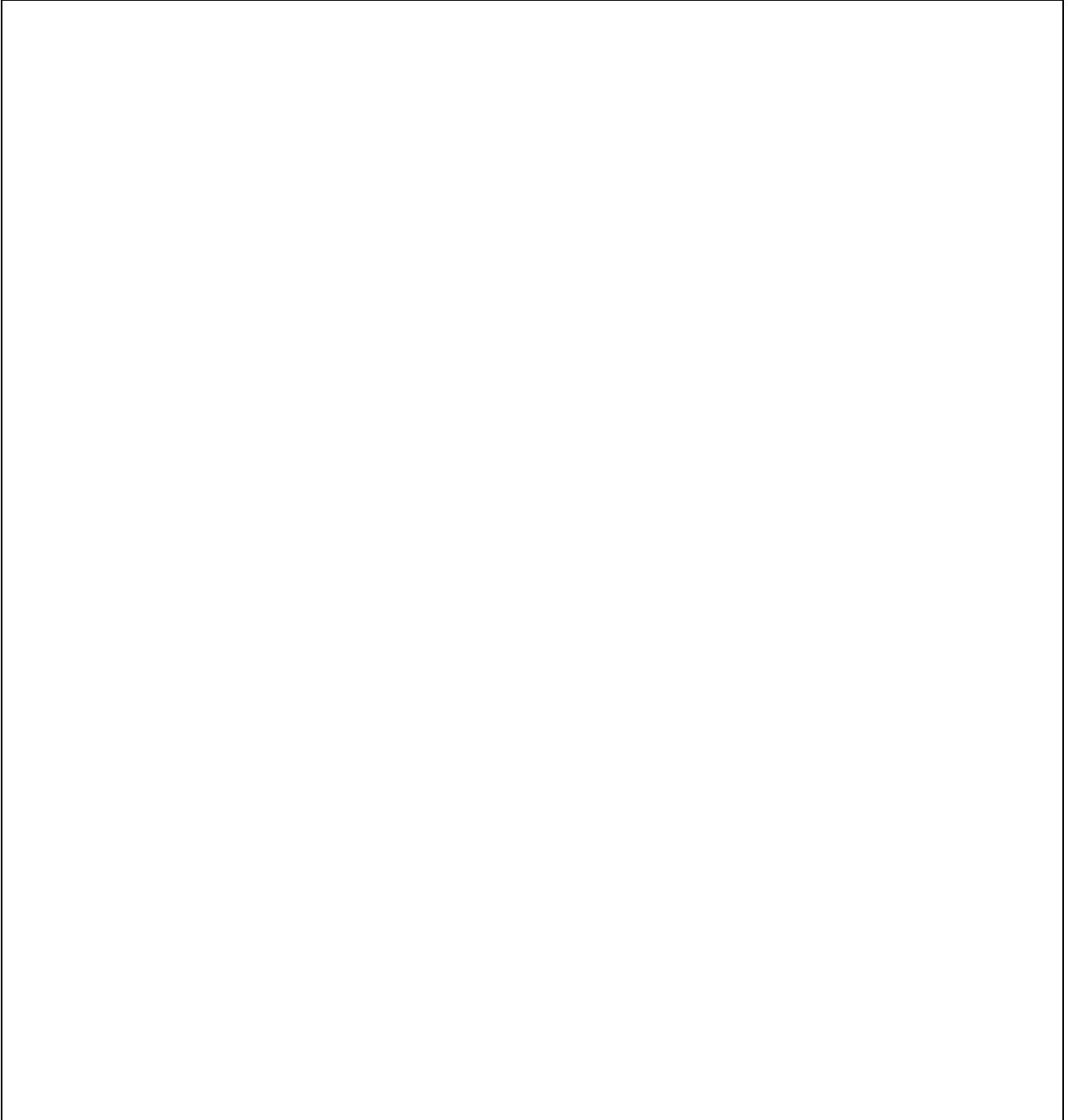
    Agnode_t *print_f = agnode(g, "Print 'f'");

    Agnode_t *end = agnode(g, "End");
    agsafeset(end, "shape", "box", "");
    agsafeset(end, "style", "rounded", "");

    Agedge_t *start_readn = aedge(g, start, read_n);
    Agedge_t *readn_printn = aedge(g, read_n, print_n);
    Agedge_t *printn_initf = aedge(g, print_n, init_f);
    Agedge_t *initsum_calc1= aedge(g, init_f, calc1);
    Agedge_t *calc1_calc2 = aedge(g, calc1, calc2);
    Agedge_t *calc2_cond1 = aedge(g, calc2, cond1);
    Agedge_t *cond1_printf = aedge(g, cond1, print_f);
    agsafeset(cond1_printf, "label", "No", "");
    Agedge_t *cond1_calc1 = aedge(g, cond1, calc1);
    agsafeset(cond1_calc1, "label", "Yes", "");
    agsafeset(cond1_calc1, "tailport", "e", "");
    agsafeset(cond1_calc1, "headport", "e", "");
    Agedge_t *printf_end = aedge(g, print_f, end);
}

```

Answer:



Question 21

Marks: 0 / 1

QE031 GraphViz-GPL-UML: Change the GraphViz program below in a way that class *Address* is connected with class *Person* (and not with class *Family*).

```
void createGraph10(Agraph_t *g) {
    Agraph_t* g_sub1 = agsubg(g, "");
    agsafeset(g_sub1, "rank", "same", "");

    Agraph_t* g_sub2 = agsubg(g, "cluster_2");
    agsafeset(g_sub2, "label", "Package Account.impl", "");
    agsafeset(g_sub2, "fontname", "Sans", "");
    agsafeset(g_sub2, "fontsize", "8", "");

    Agnode_t *family = agnode(g, "{Family|
        - surName : string\\l|
        + addPerson(Person aPerson) : void\\l
        + currentState() : double\\l
        + getAddress(): string\\l
        + getName() : string\\l
        + income (integer aAmount) : double\\l
        }");
    agsafeset(family, "shape", "record", "");
    agsafeset(family, "fontname", "Sans", "");
    agsafeset(family, "fontsize", "8", "");

    Agnode_t *address = agnode(g_sub1, "{Address|
        - street: string\\l
        - city: string\\l
        - zipCode: integer\\l
        - country: string\\l
        + getAddress(): string\\l
        }");
    agsafeset(address, "shape", "record", "");
    agsafeset(address, "fontname", "Sans", "");
    agsafeset(address, "fontsize", "8", "");

    Agnode_t *person = agnode(g_sub1, "{Person|
        - name : string\\l|
        + getName() : string\\l
        }");
    agsafeset(person, "shape", "record", "");
    agsafeset(person, "fontname", "Sans", "");
    agsafeset(person, "fontsize", "8", "");

    Agnode_t *account = agnode(g_sub1, "{BankAccount|
        - limit : double\\l
        - numberBA : integer\\l
        - value: double\\l|
        + currentValue() : double\\l
        + getAccountOwner() : integer\\l
        + getNumberBA() : integer\\l
        + getLimit() : double\\l
        }");
    agsafeset(account, "shape", "record", "");
    agsafeset(account, "fontname", "Sans", "");
    agsafeset(account, "fontsize", "8", "");

    Agnode_t *t_account = agnode(g_sub2, "{TransactionalAccount|\\l|\\l}");
    agsafeset(t_account, "shape", "record", "");
    agsafeset(t_account, "fontname", "Sans", "");
    agsafeset(t_account, "fontsize", "8", "");

    Agnode_t *saving_account = agnode(g_sub2, "{SavingAccount|\\l|\\l}");
    agsafeset(saving_account, "shape", "record", "");
    agsafeset(saving_account, "fontname", "Sans", "");
}
```

```
agsafeset(saving_account, "fontsize", "8" , "");

Agnode_t *transaction = agnode(g_sub1, "{Transaction|
    - value : integer\\l
    - date : Date\\l|
    + getAccount() : BankAccount\\l
    + getDate() : Date\\l
    + getValue() : integer\\l
    }");
agsafeset(transaction, "shape" , "record", "");
agsafeset(transaction, "fontname", "Sans" , "");
agsafeset(transaction, "fontsize", "8" , "");

Agedge_t *family_address = aedge (g, family, address);
agsafeset(family_address, "arrowhead" , "open" , "");
agsafeset(family_address, "arrowtail" , "odiamond", "");
agsafeset(family_address, "headlabel" , "1" , "");
agsafeset(family_address, "labeldistance", "1.5" , "");
agsafeset(family_address, "fontname" , "Sans" , "");
agsafeset(family_address, "fontsize" , "8" , "");

Agedge_t *family_person = aedge (g, family, person);
agsafeset(family_person, "arrowhead" , "open" , "");
agsafeset(family_person, "arrowtail" , "odiamond", "");
agsafeset(family_person, "headlabel" , "1..n" , "");
agsafeset(family_person, "labeldistance", "1.5" , "");
agsafeset(family_person, "fontname" , "Sans" , "");
agsafeset(family_person, "fontsize" , "8" , "");

Agedge_t *person_account = aedge (g, person, account);
agsafeset(person_account, "arrowhead" , "open" , "");
agsafeset(person_account, "arrowtail" , "odiamond", "");
agsafeset(person_account, "headlabel" , "1" , "");
agsafeset(person_account, "labeldistance", "1.5" , "");
agsafeset(person_account, "fontname" , "Sans" , "");
agsafeset(person_account, "fontsize" , "8" , "");
agsafeset(person_account, "minlen" , "3.0" , "");

Agedge_t *account_transaction = aedge (g, account, transaction);
agsafeset(account_transaction, "arrowhead" , "none", "");
agsafeset(account_transaction, "arrowtail" , "none", "");
agsafeset(account_transaction, "headlabel" , "0..n", "");
agsafeset(account_transaction, "taillabel" , "1" , "");
agsafeset(account_transaction, "labeldistance", "2.0" , "");
agsafeset(account_transaction, "minlen" , "3.0" , "");
agsafeset(account_transaction, "fontname" , "Sans", "");
agsafeset(account_transaction, "fontsize" , "8" , "");

Agedge_t *ta_account = aedge (g, t_account, account);
agsafeset(ta_account, "arrowhead" , "empty", "");
agsafeset(ta_account, "arrowtail" , "none" , "");

Agedge_t *account_saving = aedge (g, account, saving_account);
agsafeset(account_saving, "arrowhead", "none" , "");
agsafeset(account_saving, "arrowtail", "empty", "");
}
```

Answer:



Question 22

Marks: 0 / 1

QE032 GraphViz-GPL-FlowChart: Following flowchart diagram prints square of the biggest number. Change the GraphViz program below in a way that flowchart diagram prints the biggest number of a, b and c.

```

void createGraph11(Agraph_t *g) {
    agraphattr(g, "rankdir" , "TD" );
    agraphattr(g, "ordering" , "out");

    agnodeattr(g, "fixedsize", "true");
    agnodeattr(g, "width" , "2" );
    agnodeattr(g, "height" , "0.4" );
    agnodeattr(g, "shape" , "parallelogram");

    agedgeattr(g, "arrowhead" , "open");

    Agnode_t *start = agnode(g, "Start");
    agsafeset(start, "shape" , "box" , "");
    agsafeset(start, "style" , "rounded", "");
    agsafeset(start, "width" , "2" , "");

    Agnode_t *read_a = agnode(g, "Read a");
    Agnode_t *read_b = agnode(g, "Read b");
    Agnode_t *read_c = agnode(g, "Read c");

    Agnode_t *cond1 = agnode(g, "a>b");
    agsafeset(cond1, "shape" , "diamond", "");
    agsafeset(cond1, "height" , "0.6" , "");
    agsafeset(cond1, "width" , "1.0" , "");

    Agnode_t *cond2 = agnode(g, "a>c");
    agsafeset(cond2, "shape" , "diamond", "");
    agsafeset(cond2, "height" , "0.6" , "");
    agsafeset(cond2, "width" , "1.0" , "");

    Agnode_t *cond3 = agnode(g, "b>c");
    agsafeset(cond3, "shape" , "diamond", "");
    agsafeset(cond3, "height" , "0.6" , "");
    agsafeset(cond3, "width" , "1.0" , "");

    Agnode_t *sq_a = agnode(g, "a2 = a * a");
    agsafeset(sq_a, "shape" , "box" , "");
    agsafeset(sq_a, "width" , "2" , "");
    Agnode_t *sq_b = agnode(g, "b2 = b * b");
    agsafeset(sq_b, "shape" , "box" , "");
    agsafeset(sq_b, "width" , "2" , "");
    Agnode_t *sq_c = agnode(g, "c2 = c * c");
    agsafeset(sq_c, "shape" , "box" , "");
    agsafeset(sq_c, "width" , "2" , "");

    Agnode_t *print_a2 = agnode(g, "Print 'a2'");
    Agnode_t *print_b2 = agnode(g, "Print 'b2'");
    Agnode_t *print_c2 = agnode(g, "Print 'c2'");

    Agnode_t *end = agnode(g, "End");
    agsafeset(end, "shape" , "box" , "");
    agsafeset(end, "style" , "rounded" , "");
    agsafeset(end, "width" , "2" , "");

    Agedge_t *start_reada = agedge(g, start , read_a );
    Agedge_t *reada_readb = agedge(g, read_a, read_b );
    Agedge_t *readb_readc = agedge(g, read_b, read_c );
    Agedge_t *readc_cond1 = agedge(g, read_c, cond1 );
    Agedge_t *cond1_cond2 = agedge(g, cond1 , cond2 );
    agsafeset(cond1_cond2, "label" , "Yes" , "");

```

```
Agedge_t *cond1_cond3      = agedge(g, cond1 , cond3 );
agsafeset(cond1_cond3, "label", "No", "");

Agedge_t *cond2_printa     = agedge(g, cond2 , sq_a);
agsafeset(cond2_printa, "label", "Yes", "");
Agedge_t *cond2_printb     = agedge(g, cond2 , sq_c);
agsafeset(cond2_printb, "label", "No", "");

Agedge_t *cond3_printc     = agedge(g, cond3 , sq_c);
agsafeset(cond3_printc, "label", "No", "");
Agedge_t *cond3_printb     = agedge(g, cond3 , sq_b);
agsafeset(cond3_printb, "label", "Yes", "");

Agedge_t *sqa_print_a2     = agedge(g, sq_a, print_a2 );
Agedge_t *sqa_print_b2     = agedge(g, sq_b, print_b2 );
Agedge_t *sqa_print_c2     = agedge(g, sq_c, print_c2 );

Agedge_t *printa2_end      = agedge(g, print_a2, end );
Agedge_t *printb2_end      = agedge(g, print_b2, end );
Agedge_t *printc2_end      = agedge(g, print_c2, end );
}
```

Answer:

--

End time:

