Sample Assignment 1, 2020-09-09

Question 1 (Bitwise Operations). Write the output (and the content of variables `a,b,c` in hexadecimal notation), after this snipped is executed:

```
int a = 47;
int b = -13;
char c = 'c';
cout << (a & b) << endl;  // bitwise AND
cout << (a | b) << endl;  // bitwise OR
cout << (~b) << endl;     // bitwise NOT
```

Question 2 (Compilation and Linking).
All these command-lines are frequently used to deal with C++ code on Linux. Match the commands with their informal descriptions.

```
(1)  g++ -Wall -g -o Hello Hello.cpp
(2)  g++ -o Hello Hello.cpp; chmod a+x Hello; ./Hello
(3)  g++ -o Myprogram Myprogram.o MyprogramMain.o
(4)  g++ -o Myprogram.o -c Myprogram.cpp
(5)  ./Hello >> out.txt
```

(A) Compile and link a C++ program, then run it.

(B) Run a program, append its STDOUT (cout) to file `out.txt`

(C) Just compile a program into an object file.

(D) Compile and link a C++ program, include warnings and debug info.

(E) Just link a program from object files.

Question 3 (Makefiles).
Draw the "make" evaluation tree, if the user types `make all`
At the top of this tree specify the target `all`, then draw all all its prerequisites as children, then draw their children, if any, etc. Do this until you reach targets without any prerequisites.
Describe in English, what values are contained in `SRCFILES` and `OBJFILES`.

```
SRCFILES := $(wildcard ./*.cpp)
OBJFILES := $(patsubst ./%.cpp,./%.o,$(SRCFILES))

clean:
        rm -f hello.o hello.exe *out.txt

all: hello.exe

hello.exe: hello.o hellomain.o
        g++ -o hello.exe hello.o hellomain.o
```

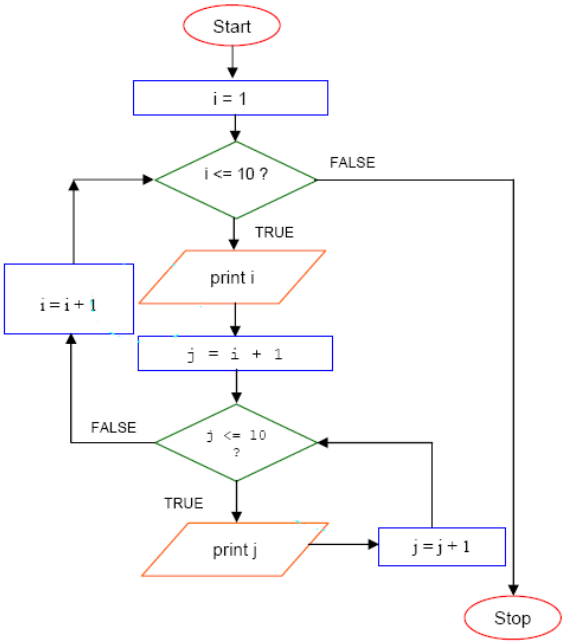
```
hello.o: hello.cpp
        g++ -c hello.cpp

hellomain.o: hellomain.cpp
        g++ -c hellomain.cpp
```

Question 4 (for loops).
Consider a regular "for" loop like this:

```
for (int i =0; i<100; i++) { /* ... */ }
```

1. Is it legal to change the loop variable `i` in the body of the loop?
2. Is it legal to use the value `i` after the loop has finished?
3. Can we omit the any of the three parts in the `for`-loop? Can we omit all 3 parts as in this loop: `for (;;) /* ... */`

Question 5 (Flowchart). Write C++ code with branch and loop statements (possibly, including "break" and "continue") to implement the flowchart shown in the picture.



Question 6 (do-while loops).
For the code snippet below draw an equivalent flowchart. Does the "continue" statement jump to the bottom of the do-while loop (and retests the condition); or does it jump to the top of the do-while loop? If you pass "null" user to the method `isLast()` the program might crash.

```
User user = userDao.GetNext();
do {
    // process some user
    user = userDao.GetNext(user.Id);
    if (user == null) { continue; }
    // do more procesing for the user.
}
while (!user.isLast())
```