Patuakhali Science and Technology University

Course CIT 111

Submitted to:

Md. Mahbubur Rahman

Submitted by:

Md. Sharafat Karim

Id: 2102024

Reg: 10151

Assignment 06

Title chapter 7 (theory)

MULTIPLE CHOICE

7.1 Which of the following looping statements end with a semi colon?

Ans: (b) do while

7.2 Which of the following control structures are used in a structured programming approach?

Ans: (d) All of these

7.3 A typical repititon control structure comprises which of the following parts?

Ans: (d) All of these

7.4 Which of the following statements can be used to immediately exit from the program?

Ans: (a) exit()

7.5 Which of the following statements skip the execution of the remaining part of the loop?

Ans: (b) continue

REVIEW QUESTIONS

7.1 (a) In a pretest loop, if the body is executed n times, the test expression is executed n+1 times. [True]

~~7.2~~ (b) The number of times a control variable is updated always equals the number of loop iterations. [True]

(c) The do....while statement first executes the loop body and then evaluate the loop control expression. [True]

(d) An exit-controlled loop is executed a ~~minimum~~ minimum of one time. [True]

(e) The three loop expressions used in a for loop header must be separated by commas. [False]

(f) while loops can be used to replace for loops without any change in the body of the loop. [False]

(g) Both the pretest loops include initialization within the statement. [False]

(h) In a for loop expression, the starting value of control variable must be less than its ending value. [False]

(i) The initialization, test condition, and increment parts may be missing in a for statement. [True]

(ii) The use of continue statement is considered as unstructured programming. [False]

7.2 (a) The sentinel-controlled loop is also known as indefinite repetition loop.

(b) In a counter-controlled loop, variable known as counter is used to count the loop operations

(c) In a exit-controlled loop, if the body is executed n times, the test condition is evaluated n times.

(d) A for loop with the no test condition is known as infinity loop.

(e) The continue statement is used to skip a part of the statements in a loop.

(f) The recurring sequence set up by a for loop can be exactly replicated by using a while loop instead.

(g) The break statement is used to exit the loop immediately and continue with the statement immediately following the loop.

7.3 Can we change the value of the control variable in for statements? If yes explain its consequences.

Ans: Yes, we can change the value of the control variable in for statements. However the consequences are it can result in unintended consequences.

7.4 What is a null statement? Explain a typical use of it.

Ans: A typical null statement is using just a semicolon. It is used with conditions or loops, when no statements are required.

7.5 Use of goto should be avoided. Explain a typical example where we find the application of goto becomes necessary.

Ans: For a more complex situation with multiple nested loop and nested conditions goto becomes necessary to exit the loop under certain circumstances. For example, if we want to create a program to print even numbers with prime number filtering and terminate the loop under certain condition, we need goto.

7.6 How would you decide the use of one of the three loops in C for a given problem?

Ans: We can use any loop. But if we need to run the statements at least once, we can go for do while. for and while can be used in the same way. For can be used instead of while to save some lines.

7.7 How do we use for loops when the number of iterations are not known?

Ans: If we donot know the number of iterations, we can run an infinity loop and break the operation when our task is finished.

7.8 Explain the operations:

(a) for (n=1; n!=10; n+=2)
    sum = sum+n;

Ans: This will assign 1 to n. And n's value will increase unless it becomes equal to zero. So it will be an infinity loop.

(b) for (n=5; n<=m; n-=1)
    sum = sum+n;

Ans: This will assign n to 5 and it will decrease with each iteration. So if n is equal to or less than m, then it will work. As the value of n will decrease over time it will also be an infinity loop.

7

(c) for (n = 1;; n = n+1)
    sum = sum + n;

Ans: Here 1 will be assigned to n and with each operation it will increase. But there is no conditional statement and thus will result in an infinity loop.

(d) for (n = 1; n <= 5;)
    sum = sum + n;

Ans: Here n's value will be assigned to 1. And the loop will continue untill n<=5. But here is no increment or decrement expression and thus will cause an infinity loop.

(e) for (n = 1; n < 5 ; n++)
    n = n - 1;

Ans: Here n's value will be also 1 at first. And it will increase with each op. iteration. At the same time it will decrease for it's statement block and will cause an ∞ loop.

7.9 (a) 4 3 2 1 0

(b) 4 3 2 1

(c) it will print 5 infinity amount of times.

(d) 108

7.10 (a) while and do.... while

Ans: While loop will first test a condition, If the condition is met then the program will enterz its block statement. Where in do.... while, first the block statement will be executed once before testing condition.

(b) while and forz

Ans: While statement needs the control variable to initialize before loop and the increment orz decrement to happen inside the block statement. But in forz loop, these ~~are~~ can be done in forz loops integrated partz.

(c) break and continue

Ans: Break will exit the loop and the program will exit from the loop automatically Where continue will cause the program to avoid the rest of the lines of the loop and to runt the loop once again.

(d) break and goto

Ans: Break will exit the loop and the program will exit from the loop automatically. But goto will send the program to an different labelled place.

(e) continue and goto

Ans: Continue will cause the program to avoid the rest of the lines of the loop and to run the loop once again. Where goto will send the program to a different labelled place.

7.11 (a) Infinity loop

(b) Infinity loop

(c) 5

(d) Infinity loop

7.12 (a) for(int i=1; i<=32; i*=2) printf("%d, ",i);

(b) for(int i=1, i<=243; i*=3) printf("%d, ",i);

(c) for(int i=-9; i<=4; i+=2) printf("%d, ",i);

(d) for (int i=-10, j=1; i>=-42; i-=(j!=1?j:2), j*=2)

printf("%d",i);

7.13 (a) int m=1;

while (m<10)

{

printf(m);

m = m+1;

}

(b) while (scanf("%d", &m)!=-1)

printf(m);

7.14 (b) do printf(m);

while (scanf("%d", &m)!=-1);

(a)
```
int m=1;
do
{
  printf(m);
  m=(m+1);
} while (m<10);
```

7.15 Ans: m=0

7.16 Ans: Infinty loop.

7.17 Ans: 1

7.18 Ans: 1

7.19 Ans: We use for(;;) to run an infinity loop. It in eauivalent to while(1). Later to exit from the loop, we can use 'break' statement.

DEBUGGING EXERCISES

7.1 (a) In the while loop after the conditional expression, semicolon is not used.

```
>        while (count != 10)
```

(b) In do.... while loop, after the while and conditional expression a semicolon is requi

```
>        while (name == 1);
```

Besides, to compare, we use "==".

(c) In do.... while loop, after do, a semicolon isn't used if there are statements.

```
>        do
```

(d) In for loops semicolon is used to separate initialization and conditional expression

```
for (x=1; x>10; x = x+1)
```

(e) After for loop we donot use semicolon if we have statements. We also use braces to make a block of statements.

```
for (; m+n<10; ++n)
{
    printf ("Hello\n");
    m = m+10;
}
```

(f) We have to use block statement if we want to put multiple lines inside a block statement,

```
for (p= 10; p>0;)
{
    p = p-1;
    printf ("%of ", p);
}
```

INTERVIEW QUESTIONS

7.1 After the execution the value of a will be 10.

7.2 The final value of n will be -1.

7.3 A do-while loop will be executed at least one time.

7.4 After execution x = 20, y = 20;

```
x = x++ + ++y;
y = ++x + y++;
```

the value of x and y will be 32 and 53. Because in a statement multiple unsequenced modification of a variable will result in an unexpected error. Here x will be incremented first and will be calculated lat

7.5 The output will be,

```
Hello world
Hello world
Hello world
Hello world
Hello world
```

7.6 The exit() statement will stop the full program. It takes an integer as an argument for exit code. 0 means successful execution

7.7 Without loop we have to write 100 printf to print 1 to 100. Or we can use goto with conditional ~~en~~ statement.

7.8 Pre-test loops are while and for. Post-te loop is do..... while.