

CMPE 108 - Experiment 4

Repetitive Structures

OBJECTIVES:

- Understand how to edit, compile and execute C computer codes.
- Understand C programming: repetitive structures

NOTES:

- You should prepare the preliminary work before coming to the laboratory session and bring soft or hard copies of the preliminary work with you.
- Before writing a computer code, you should do the following steps:
 - 1) understand and analyze the problem,
 - 2) develop an algorithm and/or flowchart,
 - 3) convert the algorithm and/or the flowchart into a C code.

PRELIMINARY WORK:

1. Trace the following code segments and show the output:

a)

```
int i = 1;
while (i++ < 5){
    printf("%d ", i);}
```

b)

```
int i = 1;
while (++i < 5){
    printf("%d ", i);}
```

Note the difference between a) and b).

c)

```
int i = 1;
while (i <= 4)
{ printf(" * \n");
  printf(" *** \n");
  printf("*****\n");
  printf(" * \n");
  printf(" * \n\n");
  i++; }
```

d)

```
int i = 1;
do
{ printf(" * \n");
  printf(" *** \n");
  printf("*****\n");
  printf(" * \n");
  printf(" * \n\n");
  cout << endl;
  i++;}
```

```
    } while(i <= 4);
```

Note the difference between c) and d).

```
e) int i;
    for (i=0; i<5; i++)
        { printf("%d ",i);}
```

Re-write this loop using while-loop and do-while loop.

```
f) int i;
    i=0;
    while (i<5) {
        printf("%d ", i);
        t++;}
```

if you delete i++; what will happen? What kind of loop you will have?

g) How can you make the for-loop to be infinite? Refer to part f

```
h) for(int i=1; i<8; i++)
    { if (i==4) break;
      printf("%d ", i);}
```

Can you re-write this code-segment without break statement?

```
i) for(int i=1; i<8; i++)
    { if (i==4) continue;
      printf("%d ", i); }
```

Can you re-write this code-segment without continue statement?

TASKS during the LAB hours:

1. Consider the following code that finds the sum of all integers between 1 and the number N:

```
#include<stdio.h>
#include<math.h>
int main()
{
    int N,i, sum=0;
    printf("Enter an integer number: ");
    scanf("%d",&N);
    for(i=1;i<=N;++i)
        sum=sum+i;
    printf("The sum=%d\n", sum);
    return 0;
}
```

a) Edit, compile and execute this code. Use the following input values for N: **10**.

- b) Modify the given code to read the value of N and find and prints the sum of the even numbers only.

Note: a number i said to be even if it can be divided by 2 without a remainder, i.e., $i\%2=0$

A sample run of the program must be as follows:

```
Program to find sum of even numbers between 1 and N
Enter an integer number N: 10
The sum of even numbers is 30.
```

- c) How can you modify part b to find the average of the even numbers only?

2. Write a C program that calculates the average of N different positive integers and prints the value of N and average on the screen using,

- (a) *while* loop structure
 (b) *do-while* loop structure.

Note: Write a separate C program for each part.

3. The GPA of a student taking 5 courses is calculated as

$$\text{GPA} = \frac{\sum_{i=1}^5 p_i * cr_i}{\sum_{i=1}^5 cr_i}$$

where cr_i and p_i are, respectively, the credit and the points of the i^{th} course. The points indicate how well a student has done in a particular course and vary depending on the letter grade received from that course. More formally, the points are calculated according to the following table:

Letter grade	Points
A	4
B	3
C	2
D	1
F	0

You are asked to write one C code to calculate the GPA of 30 students in the class. Assume that all students are taking 5 courses and the letter grade is calculated according the student's course average as

- | | |
|--------------------|----------------|
| 80 ≤ average ≤ 100 | letter grade=A |
| 70 ≤ average < 80 | letter grade=B |
| 60 ≤ average < 70 | letter grade=C |
| 50 ≤ average < 60 | letter grade=D |
| Otherwise | letter grade=F |

where, the average is computed as:

$average = 0.5 * final + 0.4 * midterm + 0.1 * lab.$

You are asked to write a C code to do the following:

- 1) For each student calculate the GPA.
- 2) Find the highest GPA, and the lowest GPA.

Note: The lab, the midterm and the final grades for each student course can be entered as inputs from the keyboard.