

## MINOR ASSIGNMENT-02

### Practical Programming with C (CSE 3544)

---

**Publish on:** 19-10-2024

**Course Outcome:** CO<sub>1</sub>

**Program Outcome:** PO<sub>1</sub>

**Submission on:** 26-10-2024

**Learning Level:** L<sub>4</sub>

---

### Problem Statement:

Experiment with selection structures (**if**, **if-else**, **if-else if-else**, **switch** statements) and repetition control structure (**for**, **while** and **do-while**) to develop applications.

### Assignment Objectives:

To learn how to use C selection structures and repetition control structure in programming and when to use each type in developing programs.

1. While the **goto** statement can be used to control program flow, it's generally recommended to use selection structures and repetition control structures. What are the advantages of these over simple **goto** statement.
2. State the purpose of the **break** statement within a switch case. Can a switch case work without it? Explain your choice with an example.
3. Describe the functions of **break** and **continue** statements within loops. Illustrate their applications in various programming contexts.
4. Under what circumstances should you employ **if-else** statements, **switch** statements, or loops? What criteria should be evaluated when selecting the appropriate control flow structure?
5. Find and explain the output of the following code snippet:

```
#include<stdio.h>
int main(){
    float x = 25.0, y=10.0;
    if(y != (x - 10.0))
        x = x - 10.0;
    else
        x = x / 2.0;
    return 0;
}
```

```
#include<stdio.h>
int main(){
    float x = 25.0, y=10.0;
    if(y < 15.0)
        if(y >= 0.0)
            x = 5 * y;
        else
            x = 2 * y;
    else
        x = 3 * y;
    return 0;
}
```

```
#include<stdio.h>
int main(){
    int i = 0;
    while (i <= 5) {
        printf("%3d %3d\n",
                i, 10 - i);
        i = i + 1;
    }
    return 0;
}
```

```
#include<stdio.h>
int main(){
    int i=1;
    while ( ){
        printf ( "%d ", i++ ) ;
        if(i>10)
            break ;
    }
    return 0;
}
```

```
#include<stdio.h>
int main() {
    int i, j,n=5;
    for(i=1, j=1; j<= n; i+= 2, j++){
        printf("%d%d\n", i, j);
    }
    return 0;
}
```

```
#include<stdio.h>
int main() {
    int count = 11;
    while (--count+1);
    printf("count down is %d \n",count);
    return 0;
}
```

```
#include<stdio.h>
int main() {
    int m, n;
    for (m = 9; m > 0; --m)
        for (n = 6; n > 1; --n)
            printf("#####\n");
    return 0;
}
```

```
#include<stdio.h>
int main(){
    int i=2;
    switch(i) {
        default: printf("Hello ");
        case 1: printf("Hello ");
        case 2:
        case 3: printf("Hello ");
    }
    return(0);
}
```

```
#include <stdio.h>
int main() {
    int i = 0 ;
    while(i++) {
        printf( "%d ",i);
        if (i > 2 )
            break ;
    }
    return ( 0);
}
```

```
#include<stdio.h>
int main() {
    int a = 10;
    if(a=0){
        printf("%d %d",
            sizeof(2.3f) , sizeof(2.3));
    }
    return(0);
}
```

6. Write a program that determines the day number (1 to 366) in a year for a date that is provided as input data. As an example, January 1, 1994, is day 1. December 31, 1993, is day 365. December 31, 1996, is day 366, since 1996 is a leap year. A year is a leap year if it is divisible by four, except that any year divisible by 100 is a leap year only if it is divisible by 400. Your program should accept the month, day, and year as integers. Include a function leap that returns 1 if called with a leap year, 0 otherwise.
7. Write a program to calculate the grade of a student using **switch** case. The program should ask the user about the marks obtained by the student and find the grade according to following rule if  $mark \geq 95$  the grade 'O', if  $81 \leq mark \leq 94$  then grade 'A', if  $71 \leq mark \leq 80$  then grade 'B', if  $61 \leq mark \leq 70$  then grade 'C', if  $51 \leq mark \leq 60$  then grade 'D', if  $40 \leq mark \leq 50$  then grade 'E', if  $mark < 40$  then grade 'F'.
8. The natural logarithm can be approximated by the following series

$$\frac{x-1}{x} + \frac{1}{2} \left( \frac{x-1}{x} \right)^2 + \frac{1}{2} \left( \frac{x-1}{x} \right)^3 + \frac{1}{2} \left( \frac{x-1}{x} \right)^4 + \dots$$

Write a program that accepts x as an input through the keyboard and calculates the sum of first nine terms of this series.

9. Design a C program to display the following pattern based on the input given by the user.

**Enter the choice of the character : G**

```
A B C D E F G F E D C B A
A B C D E F   F E D C B A
A B C D E       E D C B A
A B C D         D C B A
A B C           C B A
A B             B A
A               A
```

10. Write a program to generate the multiplication table for a given number as follows

**Enter the number > 8**

```
+-----+
|  8  16  24  32  40  48  56  64  72  80 |
|  1   2   3   4   5   6   7   8   9  10 |
|  8   8   8   8   8   8   8   8   8   8 |
+-----+
```

