

Basic C & Control Flow – Q1

(**computeGrade**) Write a C program that prints the ID and grade of each student in a class. The input contains the student IDs and their marks. The range of the marks is from 0 to 100. The relationships of the marks and grades are given below:

<u>Grade</u>	<u>Mark</u>
A	100-75
B	74-65
C	64-55
D	54-45
F	44-0

Use the sentinel value -1 for student ID to indicate the end of user input. Write the program using the switch statement.

Sample input and output sessions:

Enter Student ID:

11

Enter Mark:

56

Grade = C

Enter Student ID:

21

Enter Mark:

89

Grade = A

Enter Student ID:

31

Enter Mark:

34

Grade = F

Enter Student ID:

-1

Basic C & Control Flow – Q1

Using if-else

```
#include <stdio.h>
int main()
{
    int studentNumber = 0, mark;

    printf("Enter Student ID: \n");
    scanf("%d", &studentNumber);
    while (studentNumber != -1)
    {
        printf("Enter Mark: \n");
        scanf("%d", &mark);
        if (mark >= 75)
            printf("Grade = A\n");
        else if (mark >= 65)
            printf("Grade = B\n");
        else if (mark >= 55)
            printf("Grade = C\n");
        else if (mark >= 45)
            printf("Grade = D\n");
        else
            printf("Grade = F\n");
        printf("Enter Student ID: \n");
        scanf("%d", &studentNumber);
    }

    return 0;
}
```

Sentinel control

Note:

- Sentinel control is used for the looping construct. So the while loop is used in the code. -1 is used as the sentinel value. The loop variable is studentNumber.
- For this problem, it is more convenient to use if-else statement than the switch statement.

Using switch Basic C & Control Flow – Q1

```
#include <stdio.h>
int main() {
    int studentNumber = 0, mark;

    printf("Enter Student ID: \n");
    scanf("%d", &studentNumber);
    while (studentNumber != -1)
    {
        printf("Enter Mark: \n");
        scanf("%d", &mark);
        switch ((mark+5) / 10) {
            case 10:
            case 9:
            case 8: printf("%c\n", 'A');
                    break;
            case 7: printf("%c\n", 'B');
                    break;
            case 6: printf("%c\n", 'C');
                    break;
            case 5: printf("%c\n", 'D');
                    break;
            default: printf("%c\n", 'F');
        }
        printf("Enter Student ID: \n");
        scanf("%d", &studentNumber);
    }
    return 0;
}
```

Sentinel control

Note:

- For the expression in the switch statement, you may use $(\text{mark}+5)/10$, $(\text{mark}-5)/10$ or $(\text{mark}/5)$ – for each of the situations, you need to set the corresponding case statements accordingly.
- Note that in the expression $(\text{mark}+5)/10$, it is an integer division, and it will yield an integer value as the result.
- Note that the expression in the switch statement must be in integral type (i.e. char, integer and their expressions).

Basic C & Control Flow – Q2

(printAverage) Write a C program that reads in several lines of non-negative integer numbers, computes the average for each line and prints out the average. The value -1 in each line of user input is used to indicate the end of input for that line.

Sample input and output session:

Enter number of lines:

2

Enter line 1:

2 4 6 8 -1

Average = 5.00

Enter line 2:

1 3 5 7 9 -1

Average = 5.00

Basic C & Control Flow – Q2

```
#include <stdio.h>
int main()
{
    int total, count, lines, input;
    double average;
    int i;

    printf("\nEnter number of lines: \n");
    scanf("%d", &lines);
    for (i = 0; i < lines; i++) {
        total=0; count=0;
        printf("Enter line %d: \n", i+1);
        scanf("%d", &input);
        while (input != -1)
        {
            total += input;
            count++;
            scanf("%d", &input);
        }
        average = (double)total/(double)count;
        printf("Average = %.2f\n", average);
    }
    return 0;
}
```

Counter control

Sentinel control

Sample input and output:

Enter number of lines:

2

Enter line 1:

2 4 6 8 -1

Average = 5.00

Enter line 2:

1 3 5 7 9 -1

Average = 5.00

Note:

- Nested loop is used for the implementation.
- In the nested loop, the first loop processes each line, and for each line, the second loop⁵ processes data input of each line accordingly.

Basic C & Control Flow – Q3

(printPattern) Write a C program that accepts a positive number *height* between 1 and 10 as its parameter value, and prints a triangular pattern according to *height*. A sample input and output session when the program is called is given below.

For example, *pattern(7)* will print the pattern as shown. Note that only 1, 2 and 3 are used to generate the patterns.

Sample input and output session:

Enter the height:

7

Pattern:

1

22

333

1111

22222

333333

1111111

```
#include <stdio.h>
int main()
{
```

Basic C & Control Flow – Q3

Sample input and output:

```
    int row, col, height;
    int num = 0;
    printf("Enter the height: \n");
    scanf("%d", &height);
    printf("Pattern: \n");
```

```
    for (row = 0; row < height; row++)
    {
```

```
        for (col=0; col<row+1; col++) //print numbers
            printf("%d", num+1);
        num = (num + 1) % 3; // print up to number 3
        printf("\n");
    }
```

```
    return 0;
}
```

Enter the height:

7

Pattern:

1

22

333

1111

22222

333333

1111111

Note:

- **2-dimensional – row and column – we should use nested loop for the processing.**
- **Determine the number of rows via height.**
- **For each row, you need to print the number of times the number to be printed.**
- **When printing the number, you also need to use the modulus operator in order to limit the number to be printed. You may also use the if statement. For example:**

num = num+1; if (num==3) num=0;

Basic C & Control Flow – Q4

(computeSeries)

Write a C program that computes the value of e^x according to the following formula:

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^{10}}{10!}$$

Test Case 1:

Enter x:

0.9

Result = 2.46

Test Case 2:

Enter x:

0

Result = 1.00

Test Case 3:

Enter x:

-0.9

Result = 0.41

Basic C & Control Flow – Q4

```
#include <stdio.h>
int main()
{
    int n, denominator = 1;
    float x, result = 1.0, numerator = 1.0;

    printf("Please enter the value of x: \n");
    scanf("%f", &x);
    for (n = 1; n <= 10; n++)
    {
        denominator *= n;
        numerator *= x;
        result += numerator/denominator;
    }
    printf("Result = %.2f\n", result);
    return 0;
}
```

Please enter the value of x:

1

Result = 2.72

$$e^x = 1 + \frac{\text{numerator}}{\text{denominator}} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^{10}}{10!}$$

↑
↑
↑
.....
↑

n = 1
2
3
.....
10

Initial values: result=1.0; numerator=1.0; denominator=1;

- When n=1, result = result+num/den = 1 + (1*x)/(1*n) = 1 + x/1!
- When n=2, result = 1 + x/1! + num/den = 1+x/1! + (1*x)*x/(1!*2) = 1+x/1! + x^2/2!
- When n=3, result = (1+x/1! + x^2/2!) + (x^2*x)/(2!*3) = 1+x/1! + x^2/2! + x^3/3!
- When n=4, result = 1+x/1! + x^2/2! + x^3/3! + x^4/4!
- Etc.