

Objective Part

Compulsory

Q1. Attempt all parts and each require answer 2 – 3 lines (12*2=24)

i. What is Object Code?

Object code is produced when an interpreter or a compiler translates source code into recognizable and executable machine code.

Object code is a set of instruction codes that is understood by a computer at the lowest hardware level. Object code is usually produced by a compiler that reads some higher-level computer language source instructions and translates them into equivalent machine language instructions.

ii. What are comments and how to use them in a C program?

A comment is an explanation or description of the source code of the program. It helps a developer explain logic of the code and improves program readability. At run-time, a comment is ignored by the compiler.

There are two types of comments in C:

1) A comment that starts with a slash asterisk /* and finishes with an asterisk slash */ and you can place it anywhere in your code, on the same line or several lines.

2) Single-line Comments which uses a double slash // dedicated to comment single lines

iii. Why we use do-while loop in C?

A do while loop is similar to while loop with one exception that it executes the statements inside the body of do-while before checking the condition. Using the do-while loop, we can repeat the execution of several parts of the statements. The do-while loop is mainly used in the case where we need to execute the loop at least once.

iv. What is variable initialization and why is it important?

Variable initialization refers to the process wherein a variable is assigned an initial value before it is used in the program. Without initialization, a variable would have an unknown value, which can lead to unpredictable outputs when used in computations or other operations.

v. What is Low level language?

A low-level language is a type of programming language that contains basic instructions recognized by a computer. Unlike high-level languages used by software developers, low-level code is often cryptic and not human-readable. Two common types of low-level programming languages are assembly language and machine language.

vi. List the different types of control structures in C?

There are four types of control statements in C:

1. Decision making statements
2. Selection statements
3. Iteration statements
4. Jump statements

vii. What is function prototype?

A function prototype is simply the declaration of a function that specifies function's name, parameters and return type. It doesn't contain function body. A function prototype gives information to the compiler that the function may later be used in the program.

viii. What is syntax error?

Syntax errors are mistakes in the source code, such as misspelling of an instruction mnemonic or failure to declare a label before using it in the program.

ix. What is the purpose of struct keyword in C?

struct keyword is used to create structures in C. A structure is a user defined data type in C/C++. A structure creates a data type that can be used to group items of possibly different types into a single type.

x. What is the advantage of an array over individual variables?

When storing multiple related data, it is a good idea to use arrays. This is because arrays are named using only 1 word followed by an element number. For example: to store the 10 test results of 1 student, one can use 10 different variable names (grade1, grade2, grade3... grade10). With arrays, only 1 name is used, the rest are accessible through the index name (grade[0], grade[1], grade[2]... grade[9]).

xi. What are escape sequences?

An escape sequence in C language is a sequence of characters that doesn't represent itself when used inside string literal or character. It is composed of two or more characters starting with backslash \. For example: \n represents new line.

xii. What is linker?

In computing, a linker or link editor is a computer system program that takes one or more object files and combines them into a single executable file, library file, or another "object" file.

Subjective (4*9=36)

Q2. Write a program that takes year as input (type int) and then determines whether year is leap or not.

You can check whether a year is leap or not by using this mathematical logic:

Leap Year:

If a year is divisible by 4, 100 and 400 then it is a leap year.

If a year is divisible by 4 but not divisible by 100 then it is a leap year

Not a Leap Year:

If a year is not divisible by 4 then it is not a leap year

If a year is divisible by 4 and 100 but not divisible by 400 then it is not a leap year

```
#include <stdio.h>
int main() {
    int year;
    printf("Enter a year: ");
    scanf("%d", &year);

    // leap year if perfectly divisible by 400
    if (year % 400 == 0) {
        printf("%d is a leap year.", year);
    }
    // not a leap year if divisible by 100 but not divisible by 400
    else if (year % 100 == 0) {
        printf("%d is not a leap year.", year);
    }
    // leap year if not divisible by 100 but divisible by 4
    else if (year % 4 == 0) {
        printf("%d is a leap year.", year);
    }
    // all other years are not leap years
    else {
        printf("%d is not a leap year.", year);
    }
    return 0;
}
***~~~~~
```

Q3. Write a program that displays the following output using nested loop.

```
*  
***  
*****  
*****  
  
#include<iostream>  
using namespace std;  
int main()  
{  
    int i,j;  
    for(i=1;i<=4;i++)  
    {  
        for(j=1;j<=9;j++)  
        {  
            if(j>=6-i && j<=4+i)  
                cout<<"*";  
            else  
                cout<<" ";  
        }  
        cout<<endl;  
    }  
    return 0;  
}  
*****
```

Q4. Write a C program that copies the content of text file “Book1.txt” to other new file “Book.txt”.

```
#include<iostream>  
using namespace std;  
  
int main()  
{  
    FILE *Book1 = fopen("book1.txt", "r");  
    FILE *Book2 = fopen("book2.txt", "w");  
    char c = fgetc(Book1);  
    while (c != EOF)  
    {  
        fputc(c, Book2);  
        c = fgetc(Book1);  
    }  
    fclose(Book1);  
    fclose(Book2);  
    return 0;  
}
```

```
*****
```

Q5. Write a program that takes eight (8) elements (array) from the user. Sort the array in ascending order.

```
#include <stdio.h>

int main()
{
    int arr1[8];
    int n=8, i, j, tmp;

    printf("Input %d elements in the array :\n",n);
    for(i=0;i<n;i++)
    {
        printf("element - %d : ",i);
        scanf("%d",&arr1[i]);
    }

    for(i=0; i<n; i++)
    {
        for(j=i+1; j<n; j++)
        {
            if(arr1[j] <arr1[i])
            {
                tmp = arr1[i];
                arr1[i] = arr1[j];
                arr1[j] = tmp;
            }
        }
    }
    printf("\nElements of array in sorted ascending order:\n");
    for(i=0; i<n; i++)
    {
        printf("%d ", arr1[i]);
    }
    return 0;
}
*****
```

Q6. Write a C program to find power of a number using function recursion.

```
#include <stdio.h>
int power(int n1, int n2);
int main() {
    int base, a, result;
    printf("Enter base number: ");
    scanf("%d", &base);
    printf("Enter power number(positive integer): ");
    scanf("%d", &a);
    result = power(base, a);
    printf("%d^%d = %d", base, a, result);
    return 0;
}

int power(int base, int a) {
    if (a != 0)
```

```
        return (base * power(base, a - 1));
    else
        return 1;
}
```

Q7. Write a program that defines a structures Student to store StdId (type int), StdRoll (type string) and StdMarks (type float). In main() declare array of structure variable (size 2), input records of two BSCS students and then display the records.

```
#include<iostream>

struct Student
{
    int StdId;
    char StdRoll[20];
    float StdMarks;
};

int main()
{
    struct Student s1, s2;

    printf("\nEnter details of First Student\nEnter Roll No:");
    scanf("%s", &s1.StdRoll);
    printf("Enter Student ID:");
    scanf("%d", &s1.StdId);
    printf("Enter Marks:");
    scanf("%f", &s1.StdMarks);

    printf("\nEnter details of Second Student\nEnter Roll No:");
    scanf("%s", &s2.StdRoll);
    printf("Enter Student ID:");
    scanf("%d", &s2.StdId);
    printf("Enter Marks:");
    scanf("%f", &s2.StdMarks);
    printf("\n*****Displaying Results of First Student****\n");
    printf("Roll No: %s \n", s1.StdRoll);
    printf("Student ID: %d \n", s1.StdId);
    printf("Marks: %f \n", s1.StdMarks);

    printf("\n*****Displaying Results of Second Student****\n");
    printf("Roll No: %s \n", s2.StdRoll);
    printf("Student ID: %d \n", s2.StdId);
    printf("Marks: %f \n", s2.StdMarks);

    return 0;
}
```