Subject: Fundamental of Computer Programming 2068

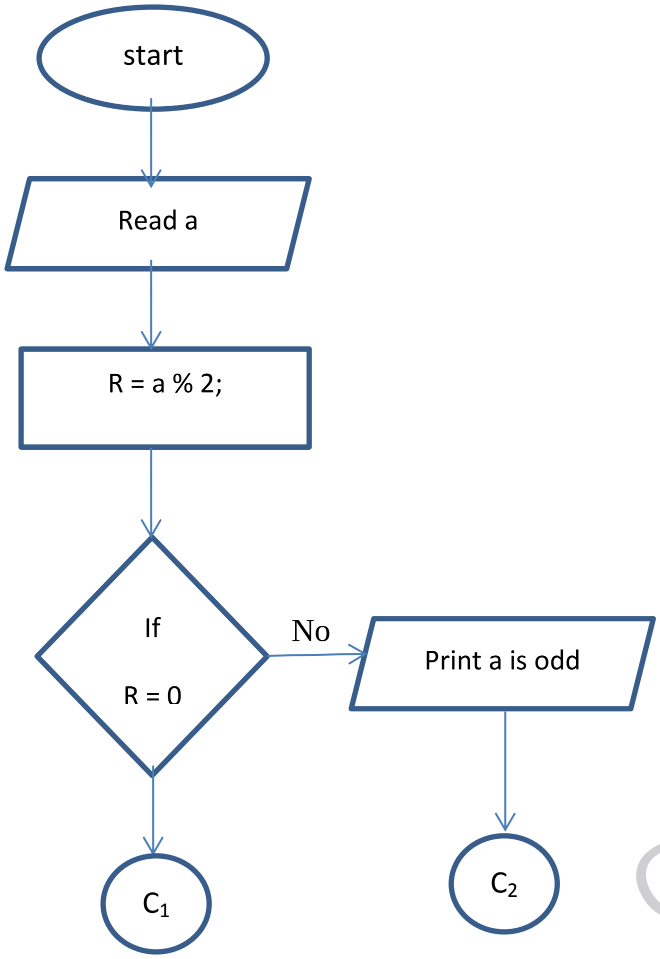
1 **Write an algorithm and flowchart to determine whether a given integer is odd or even and explain it.**

Ans:

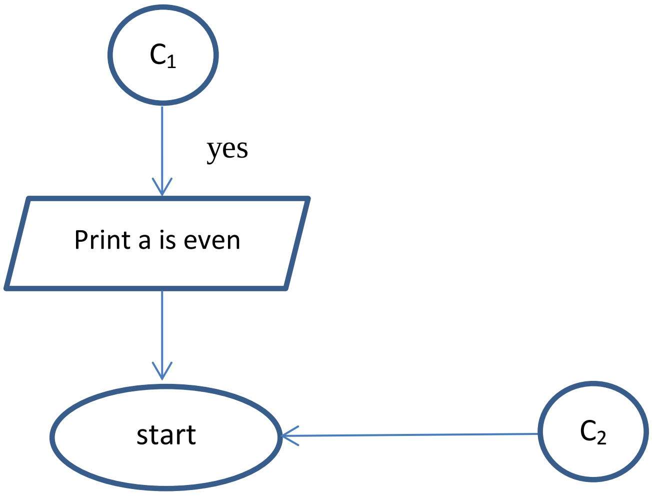
Algorithm

```
Step 1: Start
Step 2: Read a
Step 3: Find modules of a by 2 (r = a % 2)
Step 4: If r= 0
            Print a is even
            Else
            Print a is odd
Step 5: stop
```

Flowchat



Source: www.csitnepal.com



First of all, we have to read an integer a , then find the modules of a by 2 and if the value of modules is 0 then a is even else a is odd.

2. **How can you declare the variable in C ? Explain with example.**

Ans:

Variable declaration in C:

Every variable used in the program should be declared to the compiler. The decleration use two things

a. Tells the compiler the variable name
b. Specifies what type of data the variables will hold

The general format of any declaration

Data type v_1 , v_2 , v_3 , …….. , v_n ;

Where v_1,v_2,v_3,…….,v_n are variables name variables are separated by commas. The decleration statement must end with a semi colon

Example :

Int sum;

Int number , salary;

Double average , mean;

2 **Write a program to find the factorial of a given integer.**

Ans:

Program to find factorial

```
        #include<stdio.h>
        Int main()
{
        Int i = 1 , f = 1, num ;
        Printf (" enter a number :")
        Scanf(" %d",&num)
        While (i <= num)
{
        F = f*i;
        i++;
}
        Printf (" factorial of %d is %d",num,f);
        Return 0;
}
```

3 Explain switch statement with example.

Ans:

Switch statement

Switch statement allows a program to select one statement for execution out of a set of alternative. During the execution of the switch statement only one of the possible statement will be executed the remaining statement uses the simple and straight forward approach avoiding multiple uses that was in if….else statement. General format of switch statement is

```
Switch (expression)
{
        Case case1;
        Case case2;
        ………………..
```

```
    Case default
}
```

Example:

```
    #include<stdio.h>
    Void main()
{
    Int num1,num2,result;
    Char operator;
    Printf(“enter two number”);
    Scanf(“%d%d”,&num1,&num2);
    Printf(“enter an operator”);
    Scanf(“%c”,&operator);
    Switch (operator)
{
    Case ‘ + ’ :
    Result = num1 + num2
    Break;
    Case ‘ - ‘ :
    Result = num1 – num2
    Break;
    Case ‘ * ’ :
    Result = num1 * num2
    Break;
    Default:
    Printf (“unknown operator”);
    Result = 0
    Break;
}
        Printf (“%d”, result)
}
```

4 Write a program to find the largest and smallest among the given element in an array.

Ans:

```
        #include<sdtio.h>
        #include<conio.h>
        Void main()
{
        Clrscr();
        Int a[10];
        Int i , big , small;
        Printf ( “enter 10 elements in array:” );
        For ( i=0 ;I <10 ; i++ )
{
        Scanf(“  %d  ” , a[i] );
}
        Big = a[0];
        For( i=0 ; i<10 ; i++ )
        Printf( big < a[i] )
        Big a[i];
}
        Printf(“ largest element = %d ” , big );
        Small = a[0];
        For ( i=0 ; i<10 ; i++ )
{
```

```
{       If ( small > a[i] )

        Small = a[i];

}

        Printf (“ \n smallest element = %d ” , small);

}
```

5 Explain the user-define function and its types with example.

Ans:

User-define function:

The functions which are created by user for program are known as user defined functions. These type of functions are not defined in C- library and are created by the user himself. The types of user-defined functions are:

a. Functions with no arguments and no return value.
b. Functions with no arguments and a return value.
c. Functions with arguments and return value.
d. Functions with arguments and no return value.

Example:

```
        Int fact (int);

        Void main()

{

        Int n, fac;

        Clrscr ();

        Printf(“ enter the value of n ”);

        Scanf(“ %d”, &n);

        Fac = fact(n);

        Printf(“ factorial of %d is %d “, n, fac);

        Getch ();

}

        Int fact (int n)

{
        If (n==1)

        Return(1);

        Else

        Return fact (n-1) * n;

}
```

6 Write a program to accept two number and sort them with using pointer.

Ans:

```
    #include

    Void main ()

{

    Int x, y, *a, *b, temp;

    Printf (" enter the value of x and y \n ");

    Scanf (" %d %d" , &x, &y );

    Printf (" before swapping \n n = %d \n y = %d \n " ,x, y);

    a = &x;

    b = &y;

    temp = *b;

    *b = *a;

    *a = temp;

    Printf (" after swapping \n x = %d \n y = %d  \n", x, y);

}
```

7 Explain the passing structure to function with example.

Ans:

Passing structure to function:

In case of structures having numeras structure elements passing these individual elements would be hard task. In such cases we may pass whole structure to a function as shown below:

```
    #include<stdio.h>
```

```
        Structure employee
{
        Int  id;
        Char name[25];
        Float salary;
};
        Void main ()
{
        Static struct employee emp1 = { 12, “nitesh” ,7500};
        Display(emp1);
}
        Display (emp1);
{
        Struct employee emp1
{
        Printf (“ %d %s %f “, emp1, id, emp1.name, emp1.salary);
}
```

8 Write a program to accept any number and print the sum of that single digit through recursive function.

Ans:

```
#include <stdio.h>
#include<conio.h>
int sum(int);
void main() {
    int n;
    printf("Enter the single digit number");scanf("%d",&n);
    printf("Sum = %d\n", sum(n));
```

```
        getch();
}
int sum(int n)
{
    if (n <= 1)
        return 1;
    else
        return (n + sum(n-1));
}
```

9 Explain the pointer to structure with example.

Ans:

Pointer to structure:

We know the name of array stands for the address of its zero th element the same concept applies for name of arrays of structures. Suppose item is an array variable of struct type , consider the following

```
        Struct products
{
        Char name[30];
        Int manufac ;
        Float net;
}
        Item[2], ptr;
```

This statement declares item as array of two element, each type struct product and ptr as a pointer data object of type struct products.

The assignment ptr = item;

Would assign the address of zeroth element to product [0], its member can be accessed by using the following notation.

Ptr = name

Ptr = manufac

Ptr = net

OR

Write a short notes on:

a. **Dynamic memory allocation**
b. **Opening and closing file**

a) Dymanic memory allocation:

C provides the function calloc( ) and malloc( ) in the standard library and their function prototypes are in <stdlib.h> . The name calloc stands for contigeous allocation and the name malloc to dynamically create space for arrays, structures and unions. The function calloc takes to arguments where as the function malloc takes only are argument

b) Opening and closing file:

Before a program can write to a file or read from file. The program must open it , opening a file established a link between the program and the operating system, this provides the operating system, the name of the file and the mode in which the file is to be opened. The process of establishing a connection between the program and file is called opening the file.

Same is the case of closing file.