Date: 25/01/2022

EXPERIMENT NO. 3

Q. To study and Demonstrate Multi-way Decision Control Structures (Switch case)

PROGRAM STATEMENT:

a) Write a menu driven C program which will accept a 5-Digit number and depending on the user choice code will perform the following.

|  |  |
| --- | --- |
| Choice code | Action |
| A | Reverse of a number |
| B | Multiplication of the digits |
| C | Square root |
| Otherwise | Exit with Error |

b) Write a menu driven C program to perform bitwise operations on decimal number.

|  |  |
| --- | --- |
| Choice code | Action |
| A | Bitwise AND |
| B | Bitwise OR |
| C | Bitwise XOR |
| D | Bitwise NOT |
| Otherwise | Invalid Choice |

THEORY:

For all the programs listed above, we have to use switch case statement.

This is a multiple or multiway branching decision making statement. When we use nested if-else statement to check more than 1 condition then the complexity of a program increases in case of a lot of conditions. Thus, the program is difficult to read and maintain. So, to overcome this problem, C provides ‘switch case’. Switch case checks the value of an expression against a case values, if condition matches the case values then the control is transferred to that point. The case statements are executed in sequential order, but the order of the cases (including the default case) does not matter.

SYNTAX:

Switch (switch-expression)

{

case value1: statement(s)1;

case value2: statement(s)2;

break;

case value2: statement(s)2;

…..

case valueN: statement(s)N;

break;

default: statement(s)-for default;

}

Program a):

#include <stdio.h>

#include <math.h>

int main()

{

int old,number,code,rev,d1,d2,d3,d4,d5,multi;

double squart;

printf("Enter a 5-Digit number: ");

scanf("%d",&number);

old=number;

d1=number%10;

number=number/10;

d2=number%10;

number=number/10;

d3=number%10;

number=number/10;

d4=number%10;

number=number/10;

d5=number%10;

printf("1-Reverse the number, 2-Multiplication of digits,3-Sqaure root of number\n");

printf("Enter choice code: ");

scanf("%d",&code);

switch(code)

{

case 1:

rev=d1\*10000+d2\*1000+d3\*100+d4\*10+d5;

printf("Reversed number is %d",rev);

break;

case 2:

multi=d1\*d2\*d3\*d4\*d5;

printf("Multiplication of digits of number:%d",multi);

break;

case 3:

squart=sqrt(old);

printf("Sqaure root of given number is %lf",squart);

break;

default:

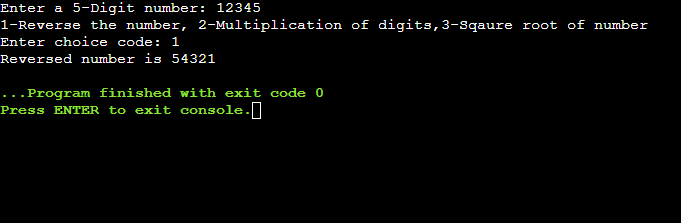
printf("Error!!!");

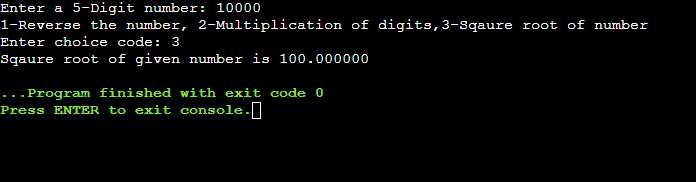
}

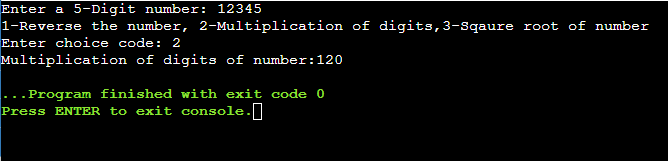
return 0;

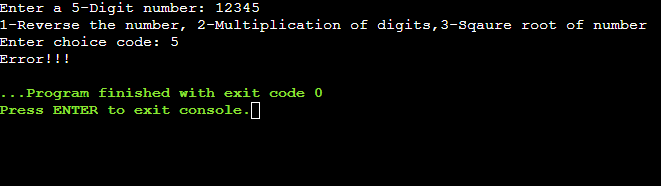
}

OUTPUT a)









Program b)

#include <stdio.h>

int main()

{

int a,b,c,d;

char code;

printf("CODES :A-Bitwise AND,B-Bitwise OR, C-Bitwise XOR,D-Bitwise NOT\n");

printf("Enter the choice code: ");

scanf("%c",&code);

printf("Enter two numbers: ");

scanf("%d" "%d",&a,&b);

switch(code)

{

case 'A':

c=a&b;

printf("a&b=%d",c);

break;

case 'B':

c=a|b;

printf("a|b=%d",c);

break;

case 'C':

c=a^b;

printf("a^b=%d",c);

break;

case 'D':

c=~a;

d=~b;

printf("~a=%d",c);

printf("\n~b=%d",d);

break;

default:

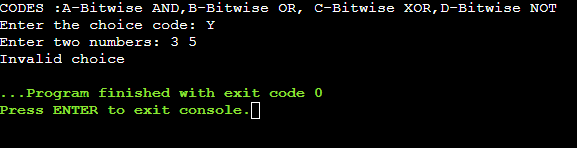
printf("Invalid choice");

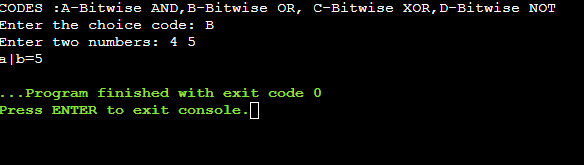
}

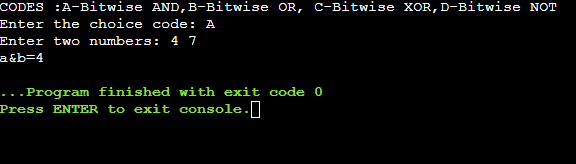
return 0;

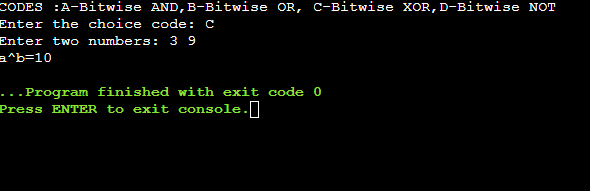
}

OUTPUT b)









CONCLUSION:

We learned about how to use switch case statements and also get to know practically how much more convenient it is in programming. However, it is good programming style to follow the logical sequence of the cases and place the default case at the end.