COMPUTER PROGRAMING

EX NO -2

Name: Kakavakam Jaswanth Sai

Roll no: CH.EN.U4CSE20130

Section: CSE-B

Subject code: 19CSE102 LAB

EX NO -2: C CONTROL STRUCTURES

A. CO-PRIMES or NOT.

Aim:

To Write a program in C to check whether a given numbers are co-primes or not using while loop.

Algorithm:

Step 1: Start
Step 2: Declare three integral variables a, b, c, hcf, i.
Step 3: Input the values of a & b. and initialize the value of i to 1.
Step 4: calculate c = (a>b)? a: b.
Step 5: while i <= a
Step 6: if (a % i == 0 && b%i == 0) then store i value to hcf.
Step 7: increment the i value by one.
Step 8: End while.
Step 9: if hcf == 1 then print the both numbers are co primes.
Step 7: Stop

Program:

```c
#include<stdio.h>
void main()
{
    int a,b,hcf;
    printf("\nEnter the two numbers :\n");
    scanf("%d\n%d",&a,&b);
    int i=1;
    while(i<=a || i<=b)
    {
        if(a%i == 0 && b%i == 0)
        {
            hcf = i;
        }
        i++;
    }
    if( hcf == 1)
    {
        printf("1");
    }
    else
    {
        printf("0");
    }

}
```

Output:

```
Enter the two numbers :
15
14

1
Process returned 2 (0x2)   exec
Press any key to continue.
```

```
Enter the two numbers :
60
45

0
Process returned 2 (0x2)   executio
Press any key to continue.
```

Result:

Thus the program to check whether a given numbers are co-primes or not using while loop with C language has been executed and verified successfully.

B. ABUNDANT or NOT.

Aim:

To Write a program in C to check whether a given number is abundant number or not using **for loop.**

Algorithm:

Step 1: Start
Step 2: Declare three integral variables a, sum, i.
Step 3: Input the values of a & b. and initialize the value of sum to 0.
Step 4: for i=0;i<a and increment the i value by one.
Step 5: if (a % i == 0) then
Step 6: calculate sum=sum+i.
Step 7: End for.
Step 8: if sum is grater than “a” then print the “a” is an abundant number.
Step 9: else print the “a” is not a abundant number.
Step 7: Stop

Program:

```c
#include<stdio.h>
void main()
{
    int a,sum=0;
    scanf("%d",&a);
    for(int i=1;i<a;i++)
    {
        if(a%i == 0)
        {
            sum=sum+i;
        }
    }
    if(sum>a)
    {
        printf("Abundunt Number");
    }
    else
    {
        printf("Not a Abundunt Number");
    }
}
```

Output:

```
12
Abundunt Number
Process returned 15 (0xF)
Press any key to continue.
```

```
86
Not a Abundunt Number
Process returned 21 (0x15)
Press any key to continue.
```

Result:

Thus the program to check whether a given number is abundant number or not using **for loop** with C language has been executed and verified successfully.

C. SUM OF 1 TO N ODD NUMBERS.

Aim:

To Write a C program to find the sum of odd numbers from 1 to n using **do while loop**

Algorithm:

Step 1: Start
Step 2: Declare three integral variables a, sum, i.
Step 3: Input the values of a & b. and initialize the value of sum to 0.
Step 4: Do
Step 5: if (i % 2 != 0) then
Step 6: calculate sum=sum+i.
Step 7: increment the i value by one.
Step 8: While (i<=a)
Step 9: print the sum.
Step 7: Stop

Program:

```
#include<stdio.h>
void main()
{
    int a,sum=0,i=1;
    scanf("%d",&a);
    do
    {
        if(i%2 != 0)
        {
            sum = sum + i;
        }
        i++;
    }while(i<=a);
    printf("\n%d",sum);
}
```

Output:

```
12

36
Process returned 3 (0x3)   execution time : 15.585 s
Press any key to continue.
```

Result:

Thus the program to find the sum of odd numbers from 1 to n using **do while loop** with C language has been executed and verified successfully.

D. SIMPLE CALCULATOR USING SWITCH CASE.

Aim:

To write a c program using **switch case** statement to output the following 2*3=6 ,2+3=5 ,4+6=10 ,4/5=0.8.

Algorithm:

Step 1: Start
Step 2: Declare three integral variables num1, num2,a.
Step 3: Input the values of num1 & num2.
Step 4: Ask the user for type of operation they need and get it as input for * = 1,+ = 2,- = 3, / = 4.
And store it to a
Step 5: by using switch case (a)
{
Case 1: calculate num1*num2 and print the value and stop.
Case 2: calculate num1+num2 and print the value and stop.
Case 3: calculate num1-num2 and print the value and stop.
Case 4: calculate num1/num2 and print the value and stop.
}

Program:

```c
#include<stdio.h>
void main()
{
    float num1,num2;
    int a;
    printf("Enter two number : \n");
    scanf("%f %f",&num1,&num2);
    printf("\n For Multiplication enter : 1 ");
    printf("\n For Addition enter : 2 ");
    printf("\n For subtraction enter : 3 ");
    printf("\n For Division enter : 4 \n");
    scanf("%d",&a);
    switch(a)
    {
        case 1 : printf("%.0f * %.0f = %.0f",num1,num2,num1*num2);
                 break;
        case 2 : printf("%.0f + %.0f = %.0f",num1,num2,num1+num2);
                 break;
        case 3 : printf("%.0f - %.0f = %.0f",num1,num2,num1-num2);
                 break;
        case 4 : printf("%.0f / %.0f = %f",num1,num2,num1/num2);
                 break;
    }
}
```

Output:

```
Enter two number :
9 4

 For Multiplication enter : 1
 For Addition enter : 2
 For subtraction enter : 3
 For Division enter : 4
4
9 / 4 = 2.250000
```

Result:

Thus the program using switch case for making simple calculator with C language has been executed and verified successfully.

E. NESTED IF – ELSE IF -ELSE.

Aim:

To Write a program in C for the given scenario **(if –else if –else)**

Assign a value to double variable cost depending on the value of integer variable distance as follows:

Distance	**Cost**
0 through 100	5.00
More than 100 but not more than 500	8.00
More than 500 but less than 1,000	10.00
1,000 or more	12.00

Algorithm:

Step 1: Start
Step 2: Declare three integral variable dis and double variable cost.
Step 3: Input the values of dis.
Step 4: if (dis>0 && dis<=100) then cost =5.00.
Step 5: else if (dis>100 && dis<=500) then cost =8.00.
Step 6: else if (dis>500 && dis<1000) then cost =10.00.
Step 7: else the cost =12.00.
Step 8: calculate the total cost = dis*cost
Step 9: print the total cost.
Step 7: Stop

Program:

```c
#include<stdio.h>
void main()
{
    float dis;
    float cost;
    printf("Enter the distance :");
    scanf("%f",&dis);
    if(dis>0 && dis<=100)
    {
        cost=5.00;
    }
    else if(dis>100 && dis<=500)
    {
      cost=8.00;
    }
    else if(dis>500 && dis<1000)
    {
        cost=10.00;
    }
    else
    {
        cost=12.00;
    }
    float TD = dis*cost;
    printf("\nTotal cost : %f",TD);
}
```

Output:

```
Enter the distance :550

Total cost : 5500.000000
Process returned 25 (0x19)   executi
Press any key to continue.
```

Result:

Thus the program for the given scenario using **if –else if –else in** C language has been executed and verified successfully