**SSN COLLEGE OF ENGINEERING (Autonomous)**

**Affiliated to Anna University**

**DEPARTMENT OF CSE**

**UCS 1211 PROGRAMMING IN C LABORATORY A2 : Modular Programming with C**

========================================================

**REGISTRATION NO: 185001112**

**NAME: PRATHYUSH S**

**CLASS: CSE-B (SEMESTER-2)**

**========================================================**

**1. Modify A1(1) to have a function CheckOddEven(num) that checks if the num is odd or even; sets a flag accordingly and return it. Use this function to find the sum of even and odd numbers in a given input of N numbers.**

#include<stdio.h>

int checkoddeven(int a);

void main()

{

int a,n,i,osum=0,esum=0,flag;

printf("Enter no. of terms:\n");

scanf("%d",&n);

printf("Enter the numbers:\n");

for(i=1;i<=n;i++)

{

scanf("%d",&a);

flag=checkoddeven(a);

if (flag==0)

esum+=a;

else

osum+=a;

}

printf("Even sum = %d\n",esum);

printf("Odd sum = %d\n",osum);

}

int checkoddeven(int a)

{

if (a%2==0)

return 0;

else

return 1;

}

Output:

cseb112@jtl-10:gcc ass21.c –o ass21

cseb112@jtl-10:./ass21

Enter no. of terms:

5

Enter the numbers:

1

2

3

4

5

Even sum = 6

Odd sum = 9

**2. Write a C function ReverseNum(num) that takes integer num and reverses its digits. Let num be passed by reference.**

#include<stdio.h>

void rev(int \*a);

void main()

{

int a;

printf("Enter a number:\n");

scanf("%d",&a);

printf("Number = %d\n",a);

rev(&a);

printf("Reverse Number = %d\n",a);

}

void rev(int \*a)

{

int r,pro=0;

while(\*a!=0)

{

r=\*a%10;

pro=(pro\*10)+r;

\*a=\*a/10;

}

\*a=pro;

}

Output:

cseb112@jtl-10:gcc ass22.c –o ass22

cseb112@jtl-10:./ass22

Enter a number:

123456

Number = 123456

Reverse Number = 654321

**3. Write a function power(X,N) that will allow a floating-point number to be raised to an integer power. Y = X N In other words, evaluate the formula where y and x are floating-point variables and n is an integer variable. Write a C program that will read in numerical values for x and n, evaluate the formula using power(X,N), and then display the calculated result.**

#include<stdio.h>

float power(float x,int n);

void main()

{

int n;

float x;

printf("Enter x:\n");

scanf("%f",&x);

printf("Enter n:\n");

scanf("%d",&n);

printf("%f\n",power(x,n));

}

float power(float x,int n)

{

if(n==0)

{

return(1);

}

else if(n<0)

{

int i;

float pro=1;

for(i=1;i<=(-1\*n);i++)

{

pro\*=1/(x);

}

return(pro);

}

else

{

int i;

float pro=1;

for(i=1;i<=n;i++)

{

pro\*=x;

}

return(pro);

}

}

Output:

cseb112@jtl-10:gcc ass23.c –o ass23

cseb112@jtl-10:./ass23

Enter x:

-7

Enter n:

5

-16807.000000

***4. Find the product of n floating point numbers. The numbers should be read from the keyboard. You should not use any looping construct. [Hint: use recursion and decide a suitable sentinel for termination of recursion.]***

#include<stdio.h>

float product(float pro);

void main()

{

float n=1;

printf("Enter the nubers.Enter 0 to stop:\n");

printf("Product = %f\n",product(n));

}

float product(float pro)

{

float a;

scanf("%f",&a);

if(a!=0)

{

pro\*=a;

product(pro);

}

else

return pro;

}

Output:

cseb112@jtl-10:gcc ass24.c –o ass24

cseb112@jtl-10:./ass24

Enter the nubers.Enter 0 to stop:

2

3

4

5

0

Product = 120.000000

**5. Write a recursive function that reads N and prints from N to 0.**

#include<stdio.h>

void print(int a);

void main()

{

int a;

printf("Enter a num:\n");

scanf("%d",&a);

print(a);

}

void print(int a)

{

if (a<=0)

printf("0");

else

{

printf("%d",a);

a=a-1;

print(a);

}

}

Output:

cseb112@jtl-10:gcc ass25.c –o ass25

cseb112@jtl-10:./ass25

Enter a num:

10

109876543210

**6. Factorials**

**The factorial of an integer n, written n!, is the product of all the integers from 1 to n inclusive. The factorial quickly becomes very large; 13! is too large to store as an integer on most computers, and 35! is too large for a floating-point variable. Your task is to find the rightmost non-zero digit of n!. (1<= n <= 100) For example, 5! = 1 \* 2 \* 3 \* 4 \* 5 = 120, so the rightmost non-zero digit of 5! is 2. Also, 7! = 1 \* 2 \* 3 \* 4 \* 5 \* 6 \* 7 = 5040, so the rightmost non-zero digit of 7! is 4.**

#include<stdio.h>

int rfact(int a);

void main()

{

int a;

int b;

printf("Enter a number:\n");

scanf("%d",&a);

if (a>0 && a<=100)

{

b=rfact(a);

printf("Right most non zero digit of %d! is %d\n",a,b);

}

else

printf("Invalid input\n");

}

int rfact(int a)

{

if (a<10)

{

int pro=1,i,r;

for(i=1;i<=a;i++)

{

pro\*=i;

}

while(pro!=0)

{

r=pro%10;

if(r>0)

{

return r;

break;

}

pro=pro/10;

}

}

else if(((a/10)%2) ==0)

{

return (6\*rfact(a/5)\*rfact(a%10)%10);

}

else

{

return (4\*rfact(a/5)\*rfact(a%10)%10);

}

}

Output:

cseb112@jtl-10:gcc ass26.c –o ass26

cseb112@jtl-10:./ass26

Enter a number:

10

Right most non zero digit of 10! is 8