|  |
| --- |
|  |
| PROGRAMMING FUNDAMENTALS |
| LAB -8  K200239 |
| ABDUL REHMAN SYED |
|  |
|  |

|  |
| --- |
|  |

Question 1:

(a) Write a program to find Factorial of a Number Using Recursion

(b)Write a program to count the digits of a given number using recursion function.

a)

Code :

#include<stdio.h>

int factorial(int n)

{

if(n==0||n==1)

return n;

else

return(n\*factorial(n-1));

}

int main()

{

int n=0,c=0;

printf("Enter the number whose factorial has to be calculated : ");

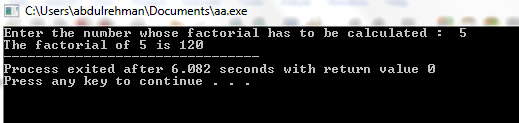
scanf("%d",&n);

c=factorial(n);

printf("The factorial of %d is %d ",n,c);

}

Output:



b)

Code:

#include<stdio.h>

int count(int n,int c)

{

if(n/10==0)

{

c++;

return c;

}

else

{

c++;

count(n/10,c);

}

}

int main()

{

int n=0,c=0,d=0;

printf("Enter the number : ");

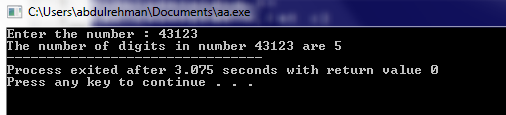
scanf("%d",&n);

d=count(n,c);

printf("The number of digits in number %d are %d",n,d);

}

Output :



Question 2:

Write a program in C to calculate the power of any number using recursion.

Test Data :

Input the base value : 2

Input the value of power : 6

Expected Output :

The value of 2 to the power of 6 is : 64

Code :

#include<stdio.h>

int pow(int n,int power,int i)

{

if(power==0)

{

return i;

}

else

{

i=n\*i;

pow(n,power-1,i);

}

}

int main()

{

int n=0,power=0,c=0;int i=1;

printf("Enter a base number : ");

scanf("%d",&n);

printf("Enter the power of exponent : ");

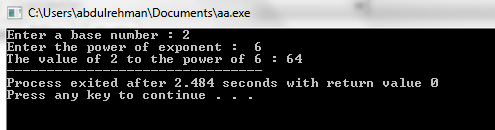
scanf("%d",&power);

c=pow(n,power,i);

printf("The value of %d to the power of %d : %d",n,power,c);

}

Output :



Question 3:

Write a program to swap two variables by passing the reference of these variables

into a function declared as void swap (int \*, int \*).

Code:

#include<stdio.h>

void swap(int \*n1,int\*n2)

{

int t=0;

t=\*n1;

\*n1=\*n2;

\*n2=t;

}

int main()

{

int n1=0,n2=0;

printf("Enter two numbers : ");

scanf("%d%d",&n1,&n2);

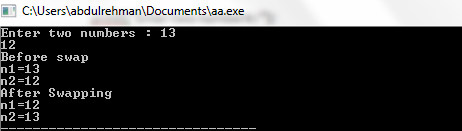
printf("Before swap\nn1=%d\nn2=%d",n1,n2);

swap(&n1,&n2);

printf("\nAfter Swapping \nn1=%d\nn2=%d",n1,n2);

}

Output :



Question 4:

1. Write a program in C to find the maximum number between two numbers

using a pointer.

2. Write a program in C to add numbers using call by reference.

1) Code:

#include<stdio.h>

int main()

{

int n1=0,n2=0;

printf("Enter the first number : ");

scanf("%d",&n1);

printf("Enter the second number : ");

scanf("%d",&n2);

int \*ptr1,\*ptr2;

ptr1=&n1;

ptr2=&n2;

if(\*ptr1>\*ptr2)

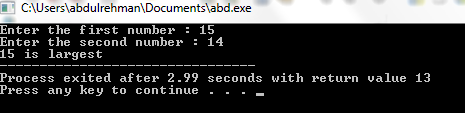
printf("%d is largest",\*ptr1);

else

printf("%d is largest",\*ptr2);

}

Output :



2)

Code :

#include<stdio.h>

int add(int \*n1,int \*n2,int \*n3)

{

\*n3=\*n1+\*n2;

}

int main()

{

int n1,n2,n3=0;

printf("Enter the first number :");

scanf("%d",&n1);

printf("Enter the second number :");

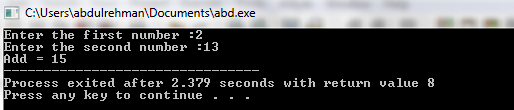
scanf("%d",&n2);

add(&n1,&n2,&n3);

printf("Add = %d",n3);

}

Output :



Question 5 :

Write a single function to calculate the square, cube and square root of its floating

point argument and make those results available to the calling program.

Code :

#include<stdio.h>

#include<math.h>

float three(float \*n,float \*s,float \*r,float \*c)

{

\*s=\*n\*\*n;

\*c=\*s\*\*n;

\*r=sqrt(\*n);

}

int main()

{

float n=0,s=0,r=0,c=0;

printf("Enter the first number : ");

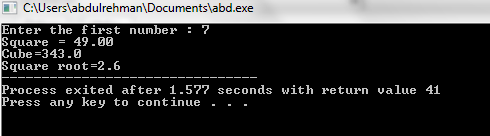
scanf("%f",&n);

three(&n,&s,&r,&c);

printf("Square = %.2f\nCube=%.1f\nSquare root=%.1f",s,c,r);

}

Output :



Question 6 :

Write a single function that receives an array of 5 integers and returns the sum,

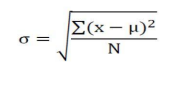
average and standard deviation of these numbers without using return statement.

Call this function from main ( ) and print the results in main ( ).

Where x represents each value in the population, μ is the mean value of the

population, Σ is the summation (or total), and N is the number of values in the

population.

Code :

#include<stdio.h>

#include<math.h>

int function(int \*i,int \*j,int \*s,float \*avg,float \*su,float \*std)

{

int q;

for(q=0;q<5;q++)

{

\*s=\*s+\*i;

i++;

}

\*avg=\*s/5;

for(q=0;q<5;q++)

{

\*su=\*su+pow(\*j-\*avg,2);

j++;

}

\*std=sqrt(\*su/5);

}

int main()

{

int i=0,a[10];float avg,su; int s=0;float std;

printf("Enter the numbers : ");

for(i=0;i<5;i++)

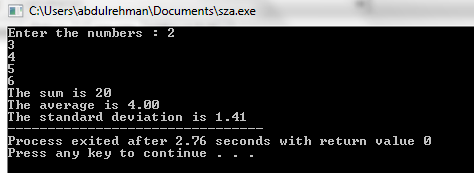
scanf("%d",&a[i]);

function(&a[0],&a[0],&s,&avg,&su,&std);

printf("The sum is %d \nThe average is %.2f \nThe standard deviation is %.2f",s,avg,std);

}

Output :



Question 7:

Write a program to store n elements in an array and print the reverse elements of an

array using pointer.

Code :

#include<stdio.h>

int rev(int \*i,int n)

{

int k=0;

for(k=0;k<n;k++)

{

i++;

}

printf("The array in reversed order : ");

for(k=0;k<n;k++)

{

i--;

printf("%d\n",\*i);

}

}

int main()

{

int n=0,i=0,a[10];

printf("Enter the limit for an array : ");

scanf("%d",&n);

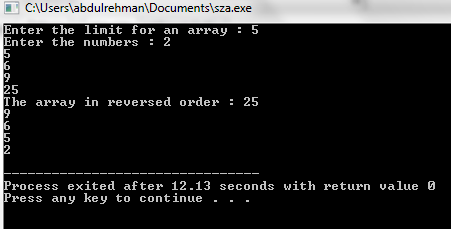
printf("Enter the numbers : ");

for(i=0;i<n;i++)

scanf("%d",&a[i]);

rev(&a[0],n);

}



Question 8 :

Suppose you have a compare() function that accepts two integers a and b.

The compare() function returns the following:

1.Return 1 if a>b

2.Return 0 if a=b

3.Return -1 if a<b

Also declare a function pointer \*fp\_func to which assign the address of the

compare( ) function. Write a c program to solve the above scenario .

Code :

#include<stdio.h>

int compare(int \*n1,int \*n2)

{

if(\*n1<\*n2)

return -1;

else if(\*n1==\*n2)

return 0;

else

return 1;

}

int main()

{

int n1=0,n2=0,c=0;

printf("Enter the first number :");

scanf("%d",&n1);

printf("Enter the second number");

scanf("%d",&n2);

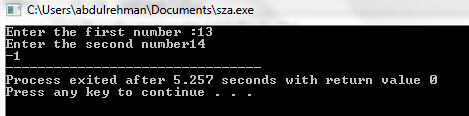
int(\*fp\_func)(int\*,int\*)=&compare;

c=fp\_func(&n1,&n2);

printf("%d",c);

}

Output :



Question 9:

Write a program to find separately the sum of the even and odd indexed elements of

an array of size 10. Pass the even and odd elements to separate functions eg:

sumEvenIndex(int\*), sumOddIndex(int\*) to carry out its sum. Also pass this array

to a function called sortArray(int[]) and display the array elements into ascending

order using pointer.

Code :

#include<stdio.h>

int sumevenindex(int \*i,int \*se)

{

int j;

for(j=0;j<10;j=j+2)

{

\*se=\*se+\*i;

i=i+2;

}

}

int sumoddindex(int \*i,int \*so)

{

int j;

i++;

for(j=1;j<10;j=j+2)

{

\*so=\*so+\*i;

i=i+2;

}

}

int sortarray(int \*ptr,int n)

{

int i=0,j=0,t=0;

for(i=0;i<n-1;i++)

{

for(j=0;j<n-i-1;j++)

{

if(\*(ptr+j)>\*(ptr+j+1))

{

t=\*(ptr+j);

\*(ptr+j)=\*(ptr+j+1);

\*(ptr+j+1)=t;

}

}

}

}

int main()

{

int a[100],se=0,so=0,i=0;

for(i=0;i<10;i++)

{

printf("%d : ",i);

scanf("%d",&a[i]);

}

sumevenindex(&a[0],&se);

printf("Sum of Even numbers : %d",se);

sumoddindex(&a[0],&so);

printf("\nSum of Odd numbers : %d",so);

sortarray(&a[0],10);

printf("\nSorted array : ");

for(i=0;i<10;i++)

{

printf("%d\n",a[i]);

}

}

Output :

