ROLL no. 1234

Name: Vivekanand Baliran Narsale.

Array:

**- Assignments on Arrays in C Programming ----------------------------------------**

1. Find all odd and even no.'s in runtime array?

#include<stdio.h>

int main()

{

int n, a[20];

printf("Enter the size of the array: ");

scanf("%d", &n);

printf("Enter array elements: \n");

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

{

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

}

printf("Even numbers are: ");

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

{

if(a[i]%2==0)

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

}

printf("Odd numbers are: ");

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

{

if(a[i]%2!=0)

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

}

return 0;

}

Output—

Enter the size of the array: 4

Enter array elements:

4

6

3

6

Even numbers are: 4 6 6 Odd numbers are: 3

--------------------------------

Process exited after 6.123 seconds with return value 0

Press any key to continue . . .

2. Find all prime no.'s in runtime array?

#include<stdio.h>

int main()

{

     int ar[20],i,n,j,counter;

     printf("Enter the size of the array ");

     scanf("%d",&n);

     printf("\nEnter array elements");

     for(i=0;i<n;i++)

     {

           scanf("%d",&ar[i]);

     }

     printf("\nPrime numbers are:");

     for(i=0;i<n;i++)

     {

           counter=0;

           for(j=2;j<ar[i];j++)

           {

                 if(ar[i]%j==0)

                 {

                       counter=1;

                       break;

                 }

           }

           if(counter==0)

           {

                 printf("\t%d",ar[i]);

           }

     }

}

Output--

Enter the size of the array 7

Enter array elements5

7

8

9

5

4

2

Prime numbers are: 5 7 5 2

--------------------------------

Process exited after 10.17 seconds with return value 0

Press any key to continue . . .

3. Write a program to create an array of integers and perform following operations on that array like  finding the sum, average, maximum and minimum number in that array. Accept the numbers of the  array from user.

#include<stdio.h>

int main()

{

     int ar[20],i,n,sum=0,average=0,max,min;

     printf("Enter the size of the array ");

     scanf("%d",&n);

     printf("\nEnter array elements");

     for(i=0;i<n;i++)

     {

           scanf("%d",&ar[i]);

     }

     for(i=0;i<n;i++){

         sum=sum+ar[i];

     }

     printf("Sum of array is:%d",sum);

     for(i=0;i<n;i++){

         average=sum/n;

     }

     printf("\nAverage of array is:%d",average);

     max = ar[0];

     for(i=0;i<n;i++){

         if(ar[i]>max){

             max=ar[i];

         }

     }

     printf("\nMaximum element of array is:%d",max);

     min= ar[0];

     for(i=0;i<n;i++){

        if(ar[i]<min){

            min=ar[i];

        }

     }

     printf("\nMinimum element of array is:%d",min);

}

Output: Enter the size of the array 5

Enter array elements4

3

5

6

7

Sum of array is:25

Average of array is:5

Maximum element of array is:7

Minimum element of array is:3

--------------------------------

Process exited after 11.46 seconds with return value 0

Press any key to continue . . .

//4. Find all odd and even no.'s in the runtime array?

#include <stdio.h>

int main() {

  int i,j,r,c, a[20][20];

  printf("Enter the number of rows: ");

  scanf("%d", &r);

  printf("Enter the number of columns: ");

  scanf("%d", &c);

  printf("\nEnter elements of matrix:\n");

  for (i = 0; i < r; ++i)

    for (j = 0; j < c; ++j) {

      printf("Enter element a [%d][%d]: ", i,j);

      scanf("%d", &a[i][j]);

    }

  printf("Even numbers are: ");

  for (i = 0; i < r; ++i)

  for (j = 0; j < c; ++j)

    {

        if(a[i][j]%2==0)

        printf("%d \t",a[i][j]);

    }

   printf("\nOdd numbers are: ");

   for (i = 0; i < r; ++i)

   for (j = 0; j < c; ++j)

    {

        if(a[i][j]%2!=0)

        printf("%d \t",a[i][j]);

    }

}

Output:

Enter the number of rows: 3

Enter the number of columns: 3

Enter elements of matrix:

Enter element a [0][0]: 3

Enter element a [0][1]: 3

Enter element a [0][2]: 4

Enter element a [1][0]: 6

Enter element a [1][1]: 3

Enter element a [1][2]: 2

Enter element a [2][0]: 5

Enter element a [2][1]: 3

Enter element a [2][2]: 2

Even numbers are: 4 6 2 2

Odd numbers are: 3 3 3 5 3

--------------------------------

Process exited after 12.42 seconds with return value 0

Press any key to continue . . .

5. Enter data for two matrices. Multiply them to store result in third

matrix & display result.

#include <stdio.h>

int main() {

  int i,j,r,c,d,k, a[20][20],b[20][20],mul[20][20];

  printf("Enter the number of rows: ");

  scanf("%d", &r);

  printf("Enter the number of columns: ");

  scanf("%d", &c);

  printf("\nEnter elements of first matrix:\n");

  for (i = 0; i < r; ++i)

    for (j = 0; j < c; ++j) {

      printf("Enter the element of a [%d][%d]: ", i,j);

      scanf("%d", &a[i][j]);

    }

  printf("\nEnter elements of second matrix:\n");

  for(i=0;i<r;i++)

   for(j=0;j<c;j++){

    printf("Enter the elements of a [%d][%d]: ",i,j);

    scanf("%d",&b[i][j]);

    }

   printf("\nMultiplication of two matrix is :\n");

   for(i=0;i<r;i++)

   for(j=0;j<c;j++){

    mul[i][j]=0;

    for(k=0;k<c;k++){

    mul[i][j]+=a[i][j]\*b[k][j];

   }

  }

   for(i=0;i<r;i++){

   for(j=0;j<c;j++){

    printf("%d  ",mul[i][j]);

   }

    printf("\n");

  }

   }

Output--

Enter elements of first matrix:

Enter the element of a [0][0]: 45

Enter the element of a [0][1]: 3

Enter the element of a [0][2]: 5

Enter the element of a [1][0]: 3

Enter the element of a [1][1]: 5

Enter the element of a [1][2]: 2

Enter the element of a [2][0]: 5

Enter the element of a [2][1]: 7

Enter the element of a [2][2]: 5

Enter elements of second matrix:

Enter the elements of a [0][0]: 4

Enter the elements of a [0][1]: 7

Enter the elements of a [0][2]: 9

Enter the elements of a [1][0]: 2

Enter the elements of a [1][1]: 3

Enter the elements of a [1][2]: 4

Enter the elements of a [2][0]: 6

Enter the elements of a [2][1]: 5

Enter the elements of a [2][2]: 4

Multiplication of two matrix is :

540 45 85

36 75 34

60 105 85

--------------------------------

Process exited after 16.14 seconds with return value 0

Press any key to continue . .

  6. Enter data for one matrix. Find its Transpose & display result.

#include <stdio.h>

int main() {

  int i,j,r,c,d,k, a[20][20],b[20][20],mul[20][20];

  printf("Enter the number of rows: ");

  scanf("%d", &r);

  printf("Enter the number of columns: ");

  scanf("%d", &c);

  printf("\nEnter elements of first matrix:\n");

  for (i = 0; i < r; ++i)

    for (j = 0; j < c; ++j) {

      printf("Enter the element of a [%d][%d]: ", i,j);

      scanf("%d", &a[i][j]);

    }

    printf("\nmatrix:\n");

    for(i=0;i<r;i++){

   for(j=0;j<c;j++){

    printf("%d  ",a[i][j]);

   }

    printf("\n");

  }

    printf("\nTransposed Matrix:\n");

    for(i=0;i<c;i++){

     for(j=0;j<r;j++){

     printf("%d  ",a[j][i]);

}

printf("\n");

}

}

Output

Enter the number of rows: 3

Enter the number of columns: 3

Enter elements of first matrix:

Enter the element of a [0][0]: 4

Enter the element of a [0][1]: 3

Enter the element of a [0][2]: 4

Enter the element of a [1][0]: 3

Enter the element of a [1][1]: 4

Enter the element of a [1][2]: 3

Enter the element of a [2][0]: 1

Enter the element of a [2][1]: 2

Enter the element of a [2][2]: 4

matrix:

4 3 4

3 4 3

1 2 4

Transposed Matrix:

4 3 1

3 4 2

4 3 4

--------------------------------

Process exited after 10.76 seconds with return value 0

Press any key to continue . . .

C program

1. Write a C program to input basic salary of an employee and calculate its Gross salary according to

following: Basic Salary <= 10000 : HRA = 20%, DA = 80% Basic Salary <= 20000 : HRA = 25%, DA = 90%

Basic Salary > 20000 : HRA = 30%, DA = 95%

#include<stdio.h>

void main4()

{

float ba,da,hra,gross;

printf("\n enter the basic salary");

scanf("%f",&ba);

if(ba<=20000)

{

da=(ba\*90)/100;

hra=(ba\*25)/100;

}else

{

if(ba<=10000)

{

da=(ba\*80)/100;

hra=(ba\*20)/100;

}else{

da=(ba\*95)/100;

hra=(ba\*30)/100;

}

}

}

Output

enter the basic salary45000

gross salary is 76500.000000

--------------------------------

Process exited after 9.688 seconds with return value 0

Press any key to continue . . .

2. Write a C program to input angles of a triangle and check whether triangle is valid or not.

#include<stdio.h>

int main()

{

int a,b,c,sum;

printf("\n Enter the tree angles of ");

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

sum=a+b+c;

if(sum==180)

{

printf("\n valid tringle exist");

}

else

{

printf("\n valid tringle not exist");

}

return 0;

}

Output

Enter the tree angles of 54

766

4

valid tringle not exist

--------------------------------

Process exited after 27.22 seconds with return value 0

Press any key to continue . . .

3. Accept a number and display it multiplication table.

#include<stdio.h>

void main()

{

int n,i=1,table;

printf("\n enter a number");

scanf("%d",&n);

while(i<=10)

{

table=i\*n;

printf("\n table is%d",table);

i++;

}

}

Output

enter a number45

table is45

table is90

table is135

table is180

table is225

table is270

table is315

table is360

table is405

table is450

--------------------------------

Process exited after 15.61 seconds with return value 0

Press any key to continue . . .

4. Accept a number and display its sum of digits.

#include<stdio.h>

void main()

{

int a,b,c,d,e,sum;

printf("\n enter a three digit number");

scanf("%d",&a);

b=a%10;

c=a/10;

d=c%10;

e=c/10;

sum=b+d+e;

printf("\n sum of digit is%d",sum);

}

Output

enter a three digit number346

sum of digit is13

--------------------------------

Process exited after 16.26 seconds with return value 0

Press any key to continue . . .

5. Accept a number and display whether its an Armstrong number.

#include<stdio.h>

int main()

{

int number,r,y=0,temp;

printf("\n enter a number");

scanf("%d",&number);

temp=number;

while(number>0)

{

r=number%10;

y=y+(r\*r\*r);

number=number/10;

}

if(temp==y)

{

printf("/n the number is armstrong number%d",y);

}

else

{

printf("\n number is not armstrog number%d",y);

}

return 0;

}

Output

enter a number345

number is not armstrog number216

--------------------------------

Process exited after 2.419 seconds with return value 0

Press any key to continue . . .

6. Write a C program to check whether a character is uppercase or lowercase alphabet.

#include<stdio.h>

#include<ctype.h>

int main()

{

char a;

printf("\n enter a character%");

scanf("%c",&a);

if(islower(a))

printf("\n the letter is small");

else

printf("\n the letter is capital");

return 0;

}

Output

enter a charactera

the letter is small

--------------------------------

Process exited after 9.376 seconds with return value 0

Press any key to continue . . .

7. Write a C program to input week number and print week day.

#include<stdio.h>

int main()

{

int week;

printf("Enter week number (1-7):");

scanf("%d",&week);

if(week==1){

printf("Monday");

}

else if(week==2){

printf("Tuesday");

}

else if(week==3){

printf("Wednesday");

}

else if(week==4){

printf("Thursday");

}

else if(week==5){

printf("Friday");

}

else if(week==6){

printf("Saturday");

}

else if(week==7){

printf("Sunday");

}

}

Output

Enter week number (1-7):6

Saturday

--------------------------------

Process exited after 11.71 seconds with return value 0

Press any key to continue . . .

8. Write a C program to input month number and print number of days in that month.

#include<stdio.h>

int main()

{

int n;

printf("\n enter the numnmber of (1-12)");

scanf("%d",&n);

if(n==1)

{

printf("\n january 31");

}

else if(n==2)

{

printf("\n feb 28/29");

}

else if(n==3)

{

printf("\n march 31");

}

else if(n==4)

{

printf("\n april 30");

}

else if(n==5)

{

printf("\n may 31");

}

else if(n==6)

{

printf("\n june 30");

}

else if(n==7)

{

printf("\n july 31");

}

else if(n==8)

{

printf("\n auguest 31");

}

else if(n==9)

{

printf("\n sept 30");

}

else if(n==10)

{

printf("\n october 31");

}

else if(n==11)

{

printf("\n nov 30");

}

else if(n==12)

{

printf("\n december 31");

}

else

{

printf("\n month is not in between 1 to 12 ");

}

}

Output

enter the numnmber of (1-12)6

june 30

--------------------------------

Process exited after 10.31 seconds with return value 0

Press any key to continue . . .

9. Write a C program to count total number of notes in given amount.

#include<stdio.h>

int main()

{

int amt;

int note1,note2,note5,note10,note20,note50,note100,note200,note500,note2000;

note1=note2=note5=note10=note20=note50=note100=note200=note500=note2000=0;

printf("Enter an amount:");

scanf("%d",&amt);

if(amt>=2000){

note2000=amt/2000;

amt-=note2000\*2000;

}

if(amt>=500){

note500=amt/500;

amt-=note500\*500;

}

if(amt>=200){

note200=amt/200;

amt-=note200\*200;

}

if(amt>=100){

note100=amt/100;

amt-=note100\*100;

}

if(amt>=50){

note50=amt/50;

amt-=note50\*50;

}

if(amt>=20){

note20=amt/20;

amt-=note20\*20;

}

if(amt>=10){

note10=amt/10;

amt-=note10\*10;

}

if(amt>=5){

note5=amt/5;

amt-=note5\*5;

}

if(amt>=2){

note2=amt/2;

amt-=note2\*2;

}

if(amt>=1){

note1=amt/1;

amt-=note1\*1;

}

printf("Number of notes:\n");

printf("2000=%d\n", note2000);

printf("500=%d\n", note500);

printf("200=%d\n", note200);

printf("100=%d\n", note100);

printf("50=%d\n", note50);

printf("20=%d\n", note20);

printf("10=%d\n", note10);

    printf("5=%d\n", note5);

    printf("2=%d\n", note2);

    printf("1=%d\n", note1);

}

Output

Enter an amount:75840

Number of notes:

2000=37

500=3

200=1

100=1

50=0

20=2

10=0

5=0

2=0

1=0

--------------------------------

Process exited after 8.56 seconds with return value 0

Press any key to continue . . .

10. Write a C program to invert the case of alphabet.

#include<stdio.h>

#include<ctype.h>

int main()

{

char a;

printf("\n enter a character");

scanf("%c",&a);

if(islower(a))

putchar(toupper(a));

else

putchar(tolower(a));

return 0;

}

OUTPUT

enter a characterV

v

--------------------------------

Process exited after 11.31 seconds with return value 0

Press any key to continue . . .

11. Write a C program to print all natural numbers from 1 to n. - using while loop

#include<stdio.h>

void main()

{

int n,count;

printf("\n Enter a upper limit");

scanf("%d",&n);

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

{

printf("\n natural numbers are:%d",count);

}

}

Output

Enter a upper limit56

natural numbers are:1

natural numbers are:2

natural numbers are:3

natural numbers are:4

natural numbers are:5

natural numbers are:6

natural numbers are:7

natural numbers are:8

natural numbers are:9

natural numbers are:10

natural numbers are:11

natural numbers are:12

natural numbers are:13

natural numbers are:14

natural numbers are:15

natural numbers are:16

natural numbers are:17

natural numbers are:18

natural numbers are:19

natural numbers are:20

natural numbers are:21

natural numbers are:22

natural numbers are:23

natural numbers are:24

natural numbers are:25

natural numbers are:26

natural numbers are:27

natural numbers are:28

natural numbers are:29

natural numbers are:30

natural numbers are:31

natural numbers are:32

natural numbers are:33

natural numbers are:34

natural numbers are:35

natural numbers are:36

natural numbers are:37

natural numbers are:38

natural numbers are:39

natural numbers are:40

natural numbers are:41

natural numbers are:42

natural numbers are:43

natural numbers are:44

natural numbers are:45

natural numbers are:46

natural numbers are:47

natural numbers are:48

natural numbers are:49

natural numbers are:50

natural numbers are:51

natural numbers are:52

natural numbers are:53

natural numbers are:54

natural numbers are:55

natural numbers are:56

--------------------------------

Process exited after 9.843 seconds with return value 0

Press any key to continue . . .

13. Write a C program to find sum of all even and odd numbers between 1 to n.

#include<stdio.h>

int main()

{

int array[20],n,i,array1[20],array2[20],j=0,k=0;

printf("\n enter the size of element");

scanf("%d",&n);

printf("\nenter the elements of array");

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

{

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

}

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

{

if(array[i]%2==0)

{

array1[j]=array[i];

j++;

}

else

{

array2[k]=array[i];

k++;

}

}

printf("\nthe even elements are ");

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

{

printf("\t %d",array1[i]);

}

printf("\nthe odd elements are");

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

{

printf("\t %d",array2[i]);

}

}

Output

enter the size of element6

enter the elements of array4

6

3

8

9

5

the even elements are 4 6 8

the odd elements are 3 9 5

--------------------------------

Process exited after 8.598 seconds with return value 0

Press any key to continue . . .

14. Write a C program to count number of digits in a number.

#include<stdio.h>

int main()

{

int n,i;

int count=0;

printf("Enter a number:");

scanf("%d",&n);

while(n!=0){

n=n/10;

count++;

}

printf("The number of digits in an integer is %d",count);

}

Output

Enter a number:456

The number of digits in an integer is 3

--------------------------------

Process exited after 8.951 seconds with return value 0

Press any key to continue . . .

15. Write a C program to find first and last digit of a number.

#include<stdio.h>

int main()

{

int n,lastdigit;

printf("Enter number:");

scanf("%d",&n);

lastdigit=n%10;

printf("Last digit of entered number is %d\n",lastdigit);

while(n>=10)

{

n=n/10;

}

printf("First digit of entered number is %d",n);

}

Output

Enter number:564

Last digit of entered number is 4

First digit of entered number is 5

--------------------------------

Process exited after 10.54 seconds with return value 0

Press any key to continue . . .

16. Write a C program to enter a number and print its reverse.

#include<stdio.h>

int main()

{

int n,rev=0,rem;

printf("Enter an integer:");

scanf("%d",&n);

while(n!=0){

rem=n%10;

rev=rev\*10+rem;

n=n/10;

}

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

}

Output

Enter an integer:456

Reversed number = 654

--------------------------------

Process exited after 7.97 seconds with return value 0

Press any key to continue . . .

17. Write a C program to check whether a number is palindrome or not.

#include<stdio.h>

int main()

{

int n,r,originalNum,sum=0;

printf("Enter a number:");

scanf("%d",&n);

originalNum=n;

while(n>0)

{

r=n%10;

sum=(sum\*10)+r;

n/=10;

}

if(originalNum==sum)

printf("%d is a palindrome number",originalNum);

else

printf("%d is not a palindrome number",originalNum);

}

Output

Enter a number:456

456 is not a palindrome number

--------------------------------

Process exited after 8.831 seconds with return value 0

Press any key to continue . . .

18. Write a C program to find power of a number using for loop.

#include<stdio.h>

void main()

{

int a,p,i=1,k,ans=1,s;

printf("enter the base number%d \n",a);

scanf("%d",&a);

printf("enter power \n ",k);

scanf("%d",&k);

while(i<=k)

{

ans=ans\*a;

i++;

}

printf(" ans is%d",ans);

}

outputenter the base number

5

enter power

6

ans is15625

--------------------------------

Process exited after 10.03 seconds with return value 0

Press any key to continue . . .

19. Write a C program to find all factors of a number.

#include<stdio.h>

int main()

{

int num,i;

printf("Enter num:");

scanf("%d",&num);

printf("Factors of %d is",num);

for(i=1;i<=num;i++){

if(num%i==0){

printf(" %d ",i);

}

}

}

Output

Enter num:54

Factors of 54 is 1 2 3 6 9 18 27 54

--------------------------------

Process exited after 10.54 seconds with return value 0

Press any key to continue . . .

20. Write a C program to calculate factorial of a number.

#include<stdio.h>

int main()

{

int num,i,fact=1;

printf("Enter a number:");

scanf("%d",&num);

for(i=1;i<=num;i++){

fact=fact\*i;

}

printf( "Factorial of %d is %d",num,fact);

}

Output

Enter a number:5

Factorial of 5 is 120

--------------------------------

Process exited after 9.085 seconds with return value 0

Press any key to continue . . .

21. Write a C program to find HCF (GCD) of two numbers.

#include<stdio.h>

int main()

{

int a,b,i,g;

printf("enter a 1st number\n");

scanf("%d",&a);

printf("enter a 2nd number\n");

scanf("%d",&b);

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

{

if(a%i==0&&b%i==0)

{

g=i;

}

}

printf("gcd/hcm is\n %d",g);

return 0;

OUTPUT

enter a 1st number

11

enter a 2nd number

9

gcd/hcm is

1

--------------------------------

Process exited after 10.31 seconds with return value 0

Press any key to continue . . .

}

22. Write a C program to find LCM of two numbers.

#include<stdio.h>

int main()

{

  int n1,n2,i,max,lcm;

    printf("Enter two integers: ");

    scanf("%d%d",&n1,&n2);

    if(n1<n2)

  max=n2;

  else

  max=n1;

    for(i=0;i<max;i++)

    {

        if(max%n1==0 && max%n2==0)

        {

            lcm = max;

            break;

    }

max++;

    }

    printf("LCM of %d and %d is %d", n1, n2, lcm);

}

Output—

Enter two integers: 54

7

LCM of 54 and 7 is 378

--------------------------------

Process exited after 11.01 seconds with return value 0

Press any key to continue . . .

23. Write a C program to check whether a number is Prime number or not.

#include<stdio.h>

int main()

{

int num,i;

int isPrime=1;

printf("Enter a number:");

scanf("%d",&num);

for(i=2;i<=num/2;i++){

if(num%i==0){

isPrime=0;

break;

}

}

if(isPrime ==1 && num>1){

printf("%d is prime number",num);

}

else{

printf("%d is not a prime number",num);

}

}

OUTPUT

Enter a number:567

567 is not a prime number

--------------------------------

Process exited after 16.48 seconds with return value 0

Press any key to continue . . .

24. Write a C program to print all Prime numbers between 1 to n.

#include<stdio.h>

int main()

{

int end,i,j,isPrime;

printf("Prime number between 1 to:");

scanf("%d",&end);

for(i=2;i<=end;i++){

isPrime=1;

for(j=2;j<=i/2;j++){

if(i%j==0){

isPrime=0;

break;

}

}

if(isPrime ==1){

printf("%d ",i);

}

}

}

Output--

Prime number between 1 to:9

2 3 5 7

--------------------------------

Process exited after 3.988 seconds with return value 0

Press any key to continue . . .

25. Write a C program to check whether a number is Perfect number or not.

#include<stdio.h>

int main()

{

int n,sum=0,i;

printf("Enter a number:");

scanf("%d",&n);

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

if(n%i==0)

{

sum=sum+i;

}

}

if(sum==n)

  printf("%d is a perfect number",n);

    else

     printf("%d is not a perfect number",n);

}

 OUTPUT

Enter a number:546

546 is not a perfect number

--------------------------------

Process exited after 13.13 seconds with return value 0

Press any key to continue . . .

26. Write a C program to check whether a number is Strong number or not.

#include <stdio.h>

void main()

{

    int i, n, k, s1=0,j,fact;

    printf("Enter a number: ");

    scanf("%d", &n);

    k = n;

    for(j=n;j>0;j=j/10)

    {

        fact = 1;

        for(i=1; i<=j % 10; i++)

        {

            fact = fact \* i;

        }

         s1 = s1 + fact;

    }

    if(s1==k)

    {

        printf("\n%d is Strong number\n", k);

    }

    else

    {

        printf("\n%d is not Strong number.", k);

    }

}

O/p:

Enter a number: 99

99 is not Strong number.

--------------------------------

Process exited after 15.12 seconds with return value 0

Press any key to continue . . .

27. Write a C program to print Fibonacci series up to n terms.

#include<stdio.h>

int main()

{

int i,c=0,n,a=0,b=1;

printf("Enter a number:");

scanf("%d",&n);

printf("Fibonacci series upto %d term is ",n);

while(c<=n){

printf("%d ",c);

a=b;

b=c;

c=a+b;

}

}

o/p

Enter a number:6

Fibonacci series upto 6 term is 0 1 1 2 3 5

--------------------------------

Process exited after 22.71 seconds with return value 0

Press any key to continue . . .

28. 28. Write a C program to print all alphabets from a to z

Write a function to generate the following pyramid of numbers

0

101

21012

3210123

432101234

54321012345

432101234

3210123

21012

101

0

#include <stdio.h>

void get();

void print(int);

int main()

{

get();

return 0;

}

void get()

{

int n;

printf("Please enter number of rows :");

scanf("%d",&n);

print(n);

}

void print(int n)

{

int i,j,k,l;

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

{

for(j = n\*2; j >= 0; j--)

if(i == j)

printf("%d",0);

else if(i > j)

printf("%d",i - j);

else if( i < j && (j - i <= i))

printf("%d",j - i);

else

printf(" ");

printf("\n");

}

for(i = n-1; i>=0; i--){

for(j = n \* 2; j>=0; j--)

if(i == j)

printf("%d",0);

else if(i > j)

printf("%d",i -j);

else if(i < j && (j-i <= i))

printf("%d", j-i);

else

printf(" ");

printf("\n");

}

}

Output

Please enter number of rows :4

0

101

21012

3210123

432101234

3210123

21012

101

0

--------------------------------

Process exited after 3.849 seconds with return value 0

Press any key to continue . . .

29. Get this output using simple loops. Here no of iterations are important.

zyxwvwxyz

zyxwxyz

zyxyz

zyz

z

#include<stdio.h>

int get();

void print(int);

int main()

{

get();

return 0;

}

int get()

{

int n;

printf("\nPlease a enter a number : ");

scanf("%d", &n);

print(n);

}

void print(int n)

{

int i,j,whitespace;

for(i=n;i>=1;i--)

{

for(whitespace=n;whitespace>i;whitespace--)

printf(" ");

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

printf("%c",123-j);

for(j=j-2;j>=1;j--)

printf("%c",123-j);

printf("\n");

}

}

Output

Please a enter a number : 5

zyxwvwxyz

zyxwxyz

zyxyz

zyz

z

--------------------------------

Process exited after 9.317 seconds with return value 0

Press any key to continue . . .

String

Q1. Write a program to sort a number of strings using bubble sort. Input is a number of strings and the output is the sorted list based on the length of strings.

#include <stdio.h>

 #include <string.h>

    int main()

    {

        char name[10][8], tname[10][8], temp[8];

        int i, j, n;

        printf("enter size: \n");

        scanf("%d", &n);

        printf("\nEnter names: ");

        for (i = 0; i < n; i++)

        {

            scanf("%s", name[i]);

            strcpy(tname[i], name[i]);

        }

        for (i = 0; i < n - 1 ; i++)

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

                if (strcmp(name[i], name[j]) > 0)

                {

                    strcpy(temp, name[i]);

                    strcpy(name[i], name[j]);

                    strcpy(name[j], temp);

                }

printf("\nResult : ");

        for (i = 0; i < n; i++)

            printf("%s   ", name[i]);

    }

Output

enter size:

5

Enter names: vivek

ram

sham

aryan

akash

Result : akash aryan ram sham vivek

--------------------------------

Process exited after 32.18 seconds with return value 0

Press any key to continue . . .

--------------------------------

Process exited after 24.24 seconds with return value 3221225477

Press any key to continue . . .

2) Define a structure to represent time in hours (0-23), minutes (0-59), and seconds (0-59), and then write a function that accepts an argument of type time represented by this structure and updates it by one second & 30 seconds.

#include <stdio.h>

typedef struct Time{

int hours;

int minutes;

int seconds;

}Time;

void updateMin(int,Time\*);

void updateSec(int,Time\*);

void setTime(Time\*);

void setHr(Time\*\*);

void setMin(Time\*\*);

void setSeconds(Time\*\*);

int main()

{

int hr = 0,mm = 0, ss = 0;

printf(" hours: ");

scanf("%d",&hr);

printf(" miutes: ");

scanf("%d",&mm);

printf(" seconds: ");

scanf("%d",&ss);

Time t1  = {hr,mm,ss};

setTime(&t1);

printf("%d %d  %d \n",t1.hours,t1.minutes,t1.seconds );

updateMin(1,&t1);

updateSec(30,&t1);

printf("%d %d  %d \n",t1.hours,t1.minutes,t1.seconds );

}

void setTime(Time \*temp)

{

setHr(&temp);

setMin(&temp);

setSeconds(&temp);

}

void setHr(Time \*\*temp)

{

int hr = (\*temp) -> hours;

if(hr >=0 && hr <24)

(\*temp) -> hours = hr;

else

(\*temp) -> hours = 0;

}

void setMin(Time \*\*temp)

{

int mm = (\*temp) -> minutes;

int r1 = 0;

if(mm >=0 && mm <60)

(\*temp)->minutes = mm;

else

{

r1 = mm/60;

if((\*temp)->hours < 24)

{

(\*temp)->hours += r1;

(\*temp)->minutes = mm % 60;

}

if((\*temp)->hours > 23)

{

(\*temp)->hours = 0;

}

}

}

void setSeconds(Time \*\*temp)

{

int ss = (\*temp) ->seconds;

int r2 = 0;

if(ss >=0 && ss < 60)

(\*temp)->seconds = ss;

else

{

r2 = ss /60;

if((\*temp)->minutes <= 59)

{

(\*temp)->minutes += r2;

(\*temp)->seconds = ss % 60;

}

if((\*temp)->minutes > 59)

{

(\*temp)->hours+=((\*temp)->minutes/60);

(\*temp)->minutes = 0;

}

if((\*temp)->hours > 23)

(\*temp)->hours = 0;

}

}

void updateMin(int mm, Time \*temp)

{

int iTemp = (temp) -> minutes;

(temp) -> minutes = iTemp+ mm;

if((temp)->minutes > 59)

setHr(&temp);

}

void updateSec(int ss, Time \*temp)

{

int iTemp = temp -> seconds;

temp -> seconds = iTemp + ss;

int iMin = (temp -> seconds) / 60;

if(temp -> seconds > 59)

updateMin(iMin,temp);

}

Output

hours: 5

miutes: 56

seconds: 45

5 56 45

5 58 75

--------------------------------

Process exited after 13.17 seconds with return value 0

Press any key to continue . . .

Q3.Using pointers write your own functions for the following:

a. String comparison b. String concatenate

c. String copy

d. String length.

#include<stdio.h>

#include<stdlib.h>

int Compare(char\*, char\*);

void Concat(char\*, char\*);

void Copy(char\*, char\*);

int Length(char\*);

int main()

{

int result = 0;

int choice = 0;

int done = 1;

char string1[20] = {'\0'};

char string2[20] = {'\0'};

while(done){

printf("\nchoose: \n");

printf("1.  comparison\n");

printf("2.  concatenate\n");

printf("3.  copy\n");

printf("4.  length\n");

printf("0. exit\n");

scanf("%d",&choice);

switch(choice){

case 1:

printf("enter: ");

scanf(" %[^'\n']s",string1);

printf("enter: ");

scanf(" %[^'\n']s",string2);

result = Compare(string1,string2);

if(result == 0)

printf("equal\n");

else

printf("not equal\n");

break;

case 2:

printf("enter: ");

scanf(" %[^'\n']s",string1);

printf("enter: ");

scanf(" %[^'\n']s",string2);

Concat(string1,string2);

printf(" Merge:  %s\n",string1);

break;

case 3:

printf("enter: ");

scanf(" %[^'\n']s",string1);

Copy(string2,string1);

printf("Copy %s: \n",string2);

break;

case 4:

printf("enter: ");

scanf(" %[^'\n']s",string1);

printf("length %d\n",Length(string1));

    break;

case 0: done = 0;

break;

}

}

}

int Compare(char \*str1, char \*str2)

{

int flag = 0;

if(str1 == NULL || str2 == NULL)

return -1;

while(( \*str1 != '\0' )&& (\*str2 != '\0'))

{

if(\*str1 !=  \*str2)

{

flag = 1;

break;

}

str1++;

str2++;

}

if(flag == 1)

return 1;

else

return 0;

}

void Concat(char \*dest, char \*src)

{

if(dest == NULL || src == NULL)

return;

while(\*dest != '\0')

dest++;

while(\*src != '\0')

{

\*dest = \*src;

dest++;

src++;

}

\*dest = '\0';

}

void Copy(char \*dest, char \*src)

{

if(dest == NULL || src == NULL)

return;

while(\*src != '\0')

{

\*dest = \*src;

dest++;

src++;

}

\*src = '\0';

}

int Length(char \*str)

{

int count = 0;

if(str == NULL)

return -1;

while(\*str != '\0')

{

str++;

count++;

}

return count;

}

Output

choose:

1. comparison

2. concatenate

3. copy

4. length

0. exit

3

enter: vivekanand

Copy vivekanand:

choose:

1. comparison

2. concatenate

3. copy

4. length

0. exit

4

enter:

narsale

length 7

choose:

1. comparison

2. concatenate

3. copy

4. length

0. exit

4: Write a program that will read each line in a file and store it

in another file with the sequence reversed, that is, the first line in file one should be  the last line in file two and so on.

Q5.Write a program to cyclically permute a string one character at a time.E.g.: If space is the input the output should produce space paces acesp cespa espac.

#include<stdio.h>

#include <string.h>

void PermuteStr(char[]);

int main()

{

char arr[20] = {'\0'};

printf("enter: ");

scanf("%[^'\n']s",arr);

PermuteStr(arr);

return 0;

}

void PermuteStr(char str[])

{

int len=strlen(str),i;

char t[len];

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

{

int j=i;

int k=0;

while(str[j]!='\0')

{

t[k]=str[j];

k++;

j++;

}

j = 0;

while(j<i)

{

t[k]=str[j];

j++;

k++;

}

printf("%s   ",t);

}

}

Output

Q6.Write functions for the following base conversion operations: a. Octal to Hexadecimal. b. Hexadecimal to Octal. Take care to validate digits/characters while accepting the input.

#include<stdio.h>

#include<conio.h>

#include<string.h>

void OcttoHex();

void HextoOct();

int main()

{

    int ch=1;

    printf("\nPlease choose 1: \n1: Octal to Hex\n2: Hex to Octal\n0:Exit\n-> ");

    scanf("%d",&ch);

    switch(ch)

    {

     case 1:

     OcttoHex();

     break;

     case 2:

     HextoOct();

     break;

     case 0:

ch=0;

break;

     default:

     printf("try again");

    }

    return 0;

}

void OcttoHex()

{

int OctNumber, Reverse=0, LastDigit, counter=0, hex=0, mul=1, i=0, k=0;

    char BinaryNumber[40] = "", HexadecimalNumber[40];

printf("Enter  Number: ");

    scanf("%d", &OctNumber);

    while(OctNumber!=0)

    {

        LastDigit = OctNumber%10;

        if(LastDigit>7)

        {

            counter++;

            break;

        }

        Reverse = (Reverse\*10) + LastDigit;

        OctNumber = OctNumber/10;

    }

    if(counter==0)

    {

        OctNumber = Reverse;

        while(OctNumber!=0)

        {

            LastDigit = OctNumber%10;

            switch(LastDigit)

            {

                case 0: strcat(BinaryNumber, "000");

                    break;

                case 1: strcat(BinaryNumber, "001");

                    break;

                case 2: strcat(BinaryNumber, "010");

                    break;

                case 3: strcat(BinaryNumber, "011");

                    break;

                case 4: strcat(BinaryNumber, "100");

                    break;

                case 5: strcat(BinaryNumber, "101");

                    break;

                case 6: strcat(BinaryNumber, "110");

                    break;

                case 7: strcat(BinaryNumber, "111");

                    break;

            }

            OctNumber = OctNumber/10;

        }

        while(BinaryNumber[k]!='\0')

            k++;

        counter=1;

        k--;

        while(k>=0)

        {

            if(BinaryNumber[k]=='0')

                LastDigit = 0;

            else

                LastDigit = 1;

            hex = hex + (LastDigit\*mul);

            if(counter%4==0)

            {

                if(hex<10)

                    HexadecimalNumber[i] = hex+48;

                else

                    HexadecimalNumber[i] = hex+55;

                mul = 1;

                hex = 0;

                counter = 1;

                i++;

            }

            else

            {

                mul = mul\*2;

                counter++;

            }

            k--;

        }

        if(counter!=1)

            HexadecimalNumber[i] = hex+48;

        if(counter==1)

            i--;

        printf("Result = ");

        counter = 0;

        for(i=i; i>=0; i--)

        {

            if(HexadecimalNumber[i]=='0' && counter==0)

            {

                counter++;

                continue;

            }

            else

                printf("%c", HexadecimalNumber[i]);

        }

        printf("\n");

    }

}

void HextoOct()

{

int i=0, chk=0, len, rem, BinDigit, OctDigit;

    long long binaryInt=0,BinNumber, temp=1, OctNumber;

    char HexArray[11], BinArray[40]="";

    printf("Enter  Number: ");

scanf("%s",HexArray);

    while(HexArray[i])

    {

        switch(HexArray[i])

        {

            case '0':

                strcat(BinArray, "0000");

                break;

            case '1':

                strcat(BinArray, "0001");

                break;

            case '2':

                strcat(BinArray, "0010");

                break;

            case '3':

                strcat(BinArray, "0011");

                break;

            case '4':

                strcat(BinArray, "0100");

                break;

            case '5':

                strcat(BinArray, "0101");

                break;

            case '6':

                strcat(BinArray, "0110");

                break;

            case '7':

                strcat(BinArray, "0111");

                break;

            case '8':

                strcat(BinArray, "1000");

                break;

            case '9':

                strcat(BinArray, "1001");

                break;

            case 'A':

                strcat(BinArray, "1010");

                break;

            case 'a':

                strcat(BinArray, "1010");

                break;

            case 'B':

                strcat(BinArray, "1011");

                break;

            case 'b':

                strcat(BinArray, "1011");

                break;

            case 'C':

                strcat(BinArray, "1100");

                break;

            case 'c':

                strcat(BinArray, "1100");

                break;

            case 'D':

                strcat(BinArray, "1101");

                break;

            case 'd':

                strcat(BinArray, "1101");

                break;

            case 'E':

                strcat(BinArray, "1110");

                break;

            case 'e':

                strcat(BinArray, "1110");

                break;

            case 'F':

                strcat(BinArray, "1111");

                break;

            case 'f':

                strcat(BinArray, "1111");

                break;

            default:

                chk = 1;

                break;

        }

        i++;

    }

    if(chk==0)

    {

        len = strlen(BinArray);

        while(len!=0)

        {

            if(BinArray[len-1]=='0')

                BinDigit=0;

            else

                BinDigit=1;

            binaryInt = binaryInt + (BinDigit\*temp);

            temp = temp\*10;

            len--;

        }

        BinNumber = binaryInt;

        OctNumber = 0;

        temp = 1;

        while(BinNumber>0)

        {

            rem = BinNumber%1000;

            switch(rem)

            {

                case 0:

                    OctDigit = 0;

                    break;

                case 1:

                    OctDigit = 1;

                    break;

                case 10:

                    OctDigit = 2;

                    break;

                case 11:

                    OctDigit = 3;

                    break;

                case 100:

                    OctDigit = 4;

                    break;

                case 101:

                    OctDigit = 5;

                    break;

                case 110:

                    OctDigit = 6;

                    break;

                case 111:

                    OctDigit = 7;

                    break;

            }

            OctNumber = (OctDigit\*temp) + OctNumber;

            BinNumber = BinNumber/1000;

            temp = temp\*10;

        }

        printf("Result = %lld", OctNumber);

    }

}

o/p

Please choose 1:

1: Octal to Hex

2: Hex to Octal

0:Exit

-> 1

Enter Number: 3

Result = 3

--------------------------------

Process exited after 48.9 seconds with return value 0

Press any key to continue . . .

Q7.A program is to be written to implement the tower of Hanoi Problem.

#include <stdio.h>

void tower(char,char,char,int);

int main()

{

int n;

printf("\nenter number: ");

scanf("%d",&n);

tower('A','B','C',n);

return 0;

}

void tower(char from,char to,char other,int n)

{

if(n==1)

printf("\n\t %c to %c",from,other);

if(n>1)

{

tower(from,other,to,n-1);

tower(from,to,other,1);

tower(to,from,other,n-1);

}

}

o/p

enter number: 4

A to B

A to C

B to C

A to B

C to A

C to B

A to B

A to C

B to C

B to A

C to A

B to C

A to B

A to C

B to C

--------------------------------

Process exited after 6.012 seconds with return value 0

Press any key to continue . . .

cpp

Q1.Create a class Person with data members as name, age, city. Write getters and setters for all the data members. Also add the display function. Create Default and Parameterized constructors. Create the object of this class in main method and invoke all the methods in that class.

#include<string.h>

 #include<iostream>

using namespace std;

 class person

 {

  int age;

string name, city;

public:

person()

  {

  age=20;

  name="name default";

  city="city default";

}

person(int age,string name,string city)

  {

  this->age=age;

  this->name=name;

  this->city=city;

}

void setAge(int age)

  {

  this->age=age;

  }

  void setName(string name)

  {

  this->name=name;

  }

  void setCity(string city)

  {

  this->city=city;

  cout<<city;

  }

  int getAge()

  {

  return age;

}

string getName()

{

return name;

}

string getCity()

{

return city;

}

//method to display values belonging to certain object

  void display()

  {

cout<<"name: "<<name<<"\nage:  "<<age<<"\nlocation: "<<city<<endl;

}

 };

 int main()

 {

  person p1;

  p1.display();

  p1.setName("name set");

  p1.getName();

  p1.setCity("city Set");

  p1.getCity();

  p1.setAge(23);

  p1.getAge();

  p1.display();

person p2(23,"name","city");

p2.display();

return 0;

 }

Output

name: name default

age: 20

location: city default

city Setname: name set

age: 23

location: city Set

name: name

age: 23

location: city

--------------------------------

Process exited after 6.824 seconds with return value 0

Press any key to continue . . .

Q2.create a class Date with data members as dd,mm,yy. Write getters and setters for all the data members. Also add the display function. Create default and parameterized constructor.

create the object of this class in main method and invoke all the methods all the methods in that class.

#include<iostream>

#include<stdio.h>

using namespace std;

class date

{

int d,m,y;

public:

date()

{

this-> d=2;

this-> m=2;

this-> y=2;

}

date(int d,int m,int y)

{

this->d=d;

this->m=m;

this->y=y;

}

void setd(int d)

{

this-> d=d;

}

void setm(int m)

{

this-> m=m;

}

void sety(int y)

{

this-> y=y;

}

int getd()

{

return d;

}

int getm()

{

return m;

}

int gety()

{

return y;

}

void display()

{

cout<<d<<"/"<< m<<"/"<< y<<endl;

}

};

int main()

{

date d1;

d1.display();

d1.setd(24);

d1.getd();

d1.setm(02);

d1.getm();

d1.sety(2022);

d1.gety();

d1.display();

date d2(12,01,1998);

d2.display();

return 0;

}

Output

0/0/0

24/02/2022

12/01/1998

--------------------------------

Process exited after 6.703 seconds with return value 0

Press any key to continue . . .

1. Create a class Book with data members as bname,id,author,price. Write getters and setters for all thedata members. Also add the display function. Create Default and Parameterized constructors. Createthe object of this class in main method and invoke all the methods in that class.

#include<stdio.h>

1. #include<iostream>
2. #include<string.h>
3. using namespace std;
4. class book
5. {
6. string bname,author;
7. int id,price;
9. public:
10. book()
11. {
12. this->id=0;
13. this->price=0;
14. this->bname="NONE";
15. this->author="NONE";
16. }
18. book(string name,string author,int idnumber,int cost)
19. {
20. this->id=idnumber;
21. this->price=cost;
22. this->bname=name;
23. this->author=author;
24. }
25. void setname(string name)
26. {
27. this->bname=name;
28. }
29. void setauthor(string author)
30. {
31. this->author=author;
32. }
33. void setid(int idnumber)
34. {
35. this->id=idnumber;
36. }
37. void setprice(int cost)
38. {
39. this->price=cost;
40. }
41. int getid()
42. {
43. return this->id;
44. }
45. int getprice()
46. {
47. return this->price;
48. }
49. string getname()
50. {
51. return this->bname;
52. }
53. string getauthor()
54. {
55. return this->author;
56. }
57. void display()
58. {
59. cout<<"bname: "<<bname<<"\nauthor: "<<author<<"\nid: "<<id<<"\ncost: "<<price<<endl;
60. }
61. };
63. int main()
64. {
65. book b;
66. b.display();
68. b.setname("too kill a mocking bird");
69. b.getname();
70. b.setauthor("sumit");
71. b.getauthor();
72. b.setid(600)
73. b.getid();
74. b.setprice(201);
75. b.getprice();
76. b.display();
78. book b2("ok","vivekanand",4564,444);
79. b2.display();
80. return 0;
81. }

Output

bname: NONE

author: NONE

id: 0

cost: 0

bname: too kill a mocking bird

author: sumit

id: 600

cost: 201

bname: ok

author: vivekanand

id: 4564

cost: 444

--------------------------------

Process exited after 8.41 seconds with return value 0

Press any key to continue . . .

Q4.create class point with data members as x,y.Write getters and setters for all the data members.  Also add the display function. Create default and parameterized constructor.

 create the object of this class in the main method and invoke all the methods in that class.

#include<iostream>

 using namespace std;

 class point

 {

  int x,y;

public:

point()

{

this->x=0;

this->y=0;

}

point(int a,int b)

{

this-> x=a;

this-> y=b;

}

void setx(int a)

{

this->x=a;

}

void sety(int b)

{

this->y=b;

}

int getx()

{

return this-> x;

}

int gety()

{

return this-> y;

}

void display()

{

cout<<x<<"."<<y<<endl;

}

 };

 int main()

 {

  point p;

  p.display();

p.setx(23);

p.getx();

p.sety(7);

p.gety();

p.display();

point p2(7,23);

p2.display();

  return 0;

 }

Output

0.0

23.7

7.23

--------------------------------

Process exited after 6.899 seconds with return value 0

Press any key to continue . . .

5. Create a class ComplexNumber with data members real, imaginary. Create Default and

Parameterized constructors. Write getters and setters for all the data members. Also add the display

function. Create the object of this class in main method and invoke all the methods in that class.

#include<iostream>

using namespace std;

class Complex{

private:

int real;

int img;

public:

~Complex(){

cout << "destructor is called";

}

Complex(){

real = 0;

img = 0;

}

Complex(int real,int img){

this -> real = real;

this -> img = img;

}

void setReal(int real){

this -> real = real;

}

void setImg(int img){

this -> img = img;

}

int getReal(){

return real;

}

int getImg(){

return img;

}

void display(){

cout << real <<"+" << img << "i" << endl;

}

void add(Complex const &c2){ //we are making read only c2 so cant manipulate value

// c1.add(c2) => c1 = c1 + c2

real = real + c2.real;

img = img + c2.img;

}

};

int main(){

Complex c1(1,2);

Complex c2;

c2.setReal(3);

c2.setImg(5);

c1.display();

c1.add(c2);

c1.display();

cout << "c2 real " << c2.getReal() << endl;

cout << "c2 IMG " << c2.getImg() << endl;

c2.display();

return 0;

}

Output

1+2i

4+7i

c2 real 3

c2 IMG 5

3+5i