Contents

1. 2018

1.1 August

power of a number (2018-08-05 18:50)

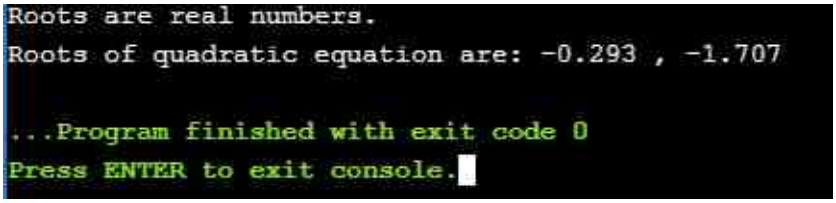
Here we are going to calculate the power of "4".

```
int main() {

int x=4;
int z= power(5,4);
printf(" %d",z);
return 0;
}
int power(int n,int y)
{
if(n==0)
{
return 1;
}
return y*power(n-1,y);
}
```

Finding the roots of a Quadratic equation (2018-08-05 22:56)

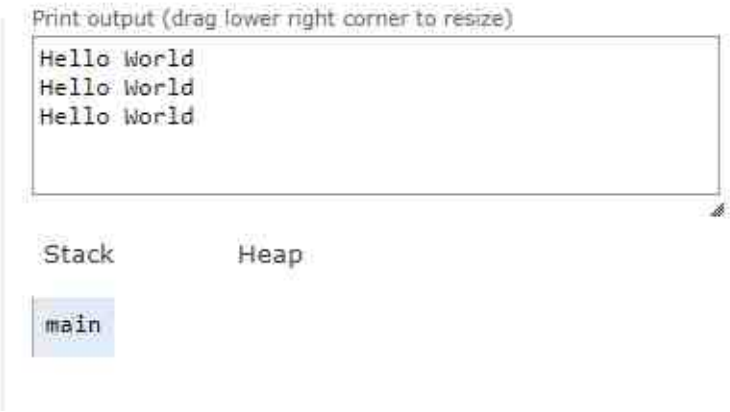
```
#include<math.h>
int main() {
float a=2,b=4,c=1;
float d,root1,root2;
d= b*b-4*a*c;
if(d<0)
{
printf("Roots are Complex\n");
printf("The roots of Quadratic expession are");
printf(" %.3f %+.3fi",-b/(2*a),sqrt(-d)/(2*a));
printf(", %.3f %+.3fi",-b/(2*a),-sqrt(-d)/(2*a));

return 0;
}
else if(d==0) {
printf("Both roots are equal.\n");
```

```
root1 = -b /(2* a);
printf("Root of quadratic equation is: %.3f ",root1);

return 0;
}
else {
printf("Roots are real numbers.\n");

root1 = ( -b + sqrt(d)) / (2* a);
root2 = ( -b - sqrt(d)) / (2* a);
printf("Roots of quadratic equation are: %.3f , %.3f",root1,root2);
}

return 0;
}
```



Print "Hello World" without using semicolon. **(2018-08-05 23:34)**

```
1. int main()
2. {

1. //using if loop
2. if(printf("Hello World")) {
3. }

//using switch
switch(printf("Hello World")) {
}
//using while
while(!printf("Hello World")) {
}
return 0;
}
```



Biggest Of Three Numbers Using Conditional operator/Ternary Operator in C (2018-08-05 23:55)

```
int main()
{
int a,b,c,max;
printf("Enter three numbers\n");
scanf(" %d %d %d", &a, &b, &c);
max = a>b?(a>c?a:c):(b>c?b:c);
printf(" %d",max);
return 0;
}
```

---

finding how many times a sorted array is rotated (2018-08-13 12:23)

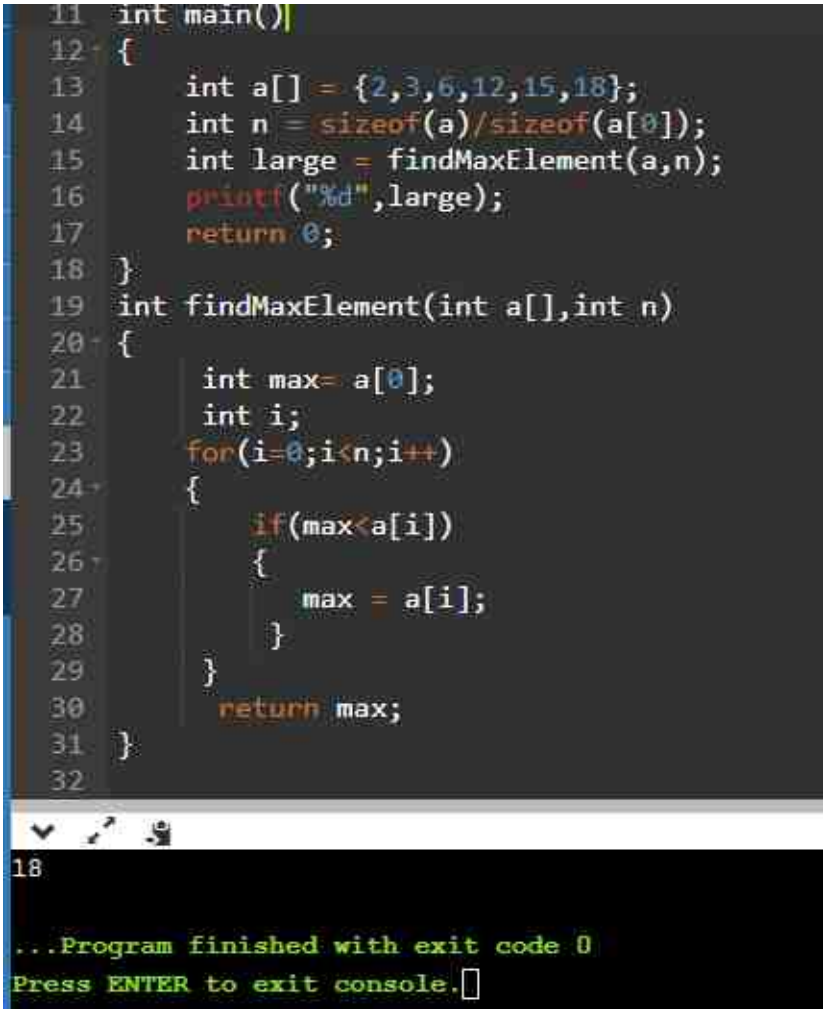
First we should have a sorted array in order to compare with the input what we have given so we can know how many times it is sorted.

```
a[] = {2,3,6,12,15,18 }.
int main()
{
int a[]= {15,18,2,3,6,12 };
int n = sizeof(a)/sizeof(a[0]);
int count = countRotations(a,n);
printf(" %d",count);
return 0;
}
int countRotations(int a[],int n)
{
int min = a[0],x;
for(int i=0;i<n;i++)
{
if(min>a[i])
{
min = a[i];
x = i;
}
}
return x;
}
```

Finding the maximum element of an array (2018-08-13 13:15)

```
#include <stdio.h>
int findMaxElement(int[], int);
int main()
{
int a[] = {2,3,6,12,15,18 };
int n = sizeof(a)/sizeof(a[0]);
int large = findMaxElement(a,n);
```

```
printf(" %d",large);
return 0;
}
int findMaxElement(int a[],int n)
{
int max= a[0];
int i;
for(i=0;i<n;i++)
{
if(max<a[i])
{
max = a[i];
}
}
return max;
}
```

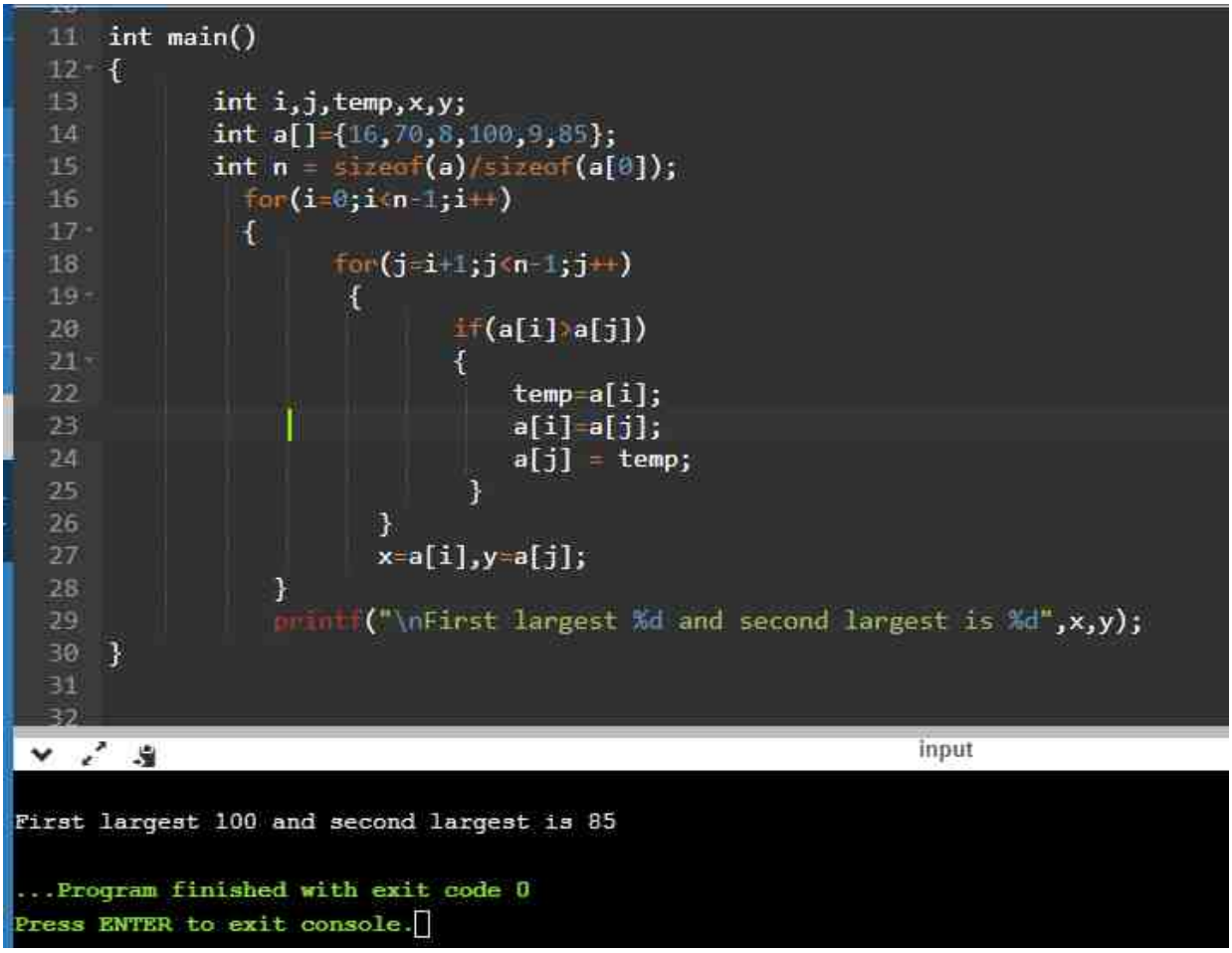


Finding the smallest and the largest element in an array (2018-08-13 16:43)

```
#include <stdio.h>
void main()
{
int i;
int a[]= {2,3,6,12,15,18 };
int n = sizeof(a)/sizeof(a[0]);
int small = a[0],max=a[0];
for(i=0;i<n;i++)
{
if(a[i]>max)
max = a[i];
if(a[i]<small)
small = a[i];
}
printf("small is %d and large is %d",small,max);
}
```

Finding the second largest in an array (2018-08-13 22:35)

```
int main()
{
int i,j,temp,x,y;
int a[]= {16,70,8,100,9,85 };
int n = sizeof(a)/sizeof(a[0]);
for(i=0;i<n-1;i++)
{
for(j=i+1;j<n-1;j++)
{
if(a[i]>a[j])
{
temp=a[i];
a[i]=a[j];
a[j] = temp;
}
}
x=a[i],y=a[j];
}
printf("\nFirst largest %d and second largest is %d",x,y);
}
```



```
int main()
{
        int i,j,temp,x,y;
        int a[]={16,70,8,100,9,85};
        int n = sizeof(a)/sizeof(a[0]);
          for(i=0;i<n-1;i++)
          {
                for(j=i+1;j<n-1;j++)
                 {
                        if(a[i]>a[j])
                        {
                            temp=a[i];
                            a[i]=a[j];
                            a[j] = temp;
                         }
                    }
                    x=a[i],y=a[j];
            }
            printf("\nFirst largest %d and second largest is %d",x,y);
}
```

input

```
First largest 100 and second largest is 85

...Program finished with exit code 0
Press ENTER to exit console.
```

---

finding first smallest and the second smallest **(2018-08-13 22:42)**

```
int main()
{
int i,j,temp,x,y;
int a[]= {16,70,8,100,9,85 };
int n = sizeof(a)/sizeof(a[0]);
for(i=0;i<n-1;i++)
{
for(j=i+1;j<n-1;j++)
{
if(a[i]<a[j])
{
temp=a[i];
a[i]=a[j];
a[j] = temp;
}
}
```

x=a[i],y=a[i-1];
}
printf("\nFirst smallest %d and second smallest is %d",x,y);
}

Array of Structures **(2018-08-15 16:06)**

#include<stdio.h>
#include<string.h>
#define MAX 3
struct student
{
char name[20];
int roll _no,i;
float marks;
};

int main()
{
struct student arr _student[MAX];
int i;

```
for(i=0;i<MAX;i++)
{
printf("\nEnter the details of student %d\n\n",i+1);

printf("Enter Name: ");
scanf(" %s", &arr _student[i].name);

printf("Enter roll no: ");
scanf(" %d", &arr _student[i].roll _no);

printf("Enter marks: ");
scanf(" %f", &arr _student[i].marks);
}
printf("\n");


printf("Name\tRollno\tMarks\n") ;
for(i=0;i<MAX;i++)
{
printf(" %s\t %d\t\t %.2f\n",arr _student[i].name,arr _student[i].roll _no,arr _student[i].marks);
}

return 0;
}
```

```
Result...
CPU Time: 0.00 sec(s), Memory: 1500 kilobyte(s)

Enter the details of student1

Enter Name: Enter roll no: Enter marks:
Enter the details of student2

Enter Name: Enter roll no: Enter marks:
Name    Rollno  Marks
Raghava 55      22.10
Abhi    66      22.10
```

Rotating the elements of an array data structure by k positions to the right. **(2018-08-15 18:41)**

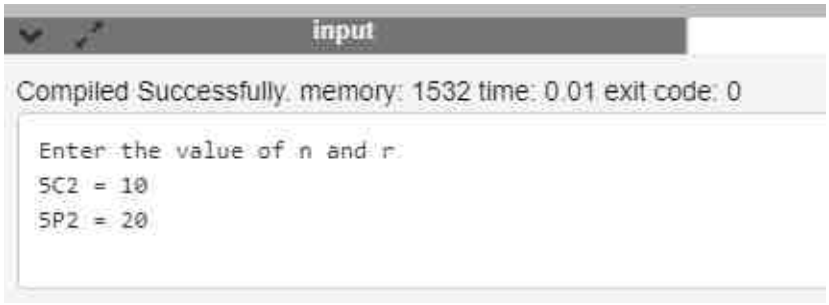
```
#include <stdio.h>
// K is no.of positions an array to be rotated
int main()
{
```

```
int k=2 , n=4,a[4]= {1,2,3,4 }, b[4],i;
for(i=0;i<n;i++)
{
b[(i+k) %n]=a[i];
}
for(i=0;i<n;i++)
{
printf(" %d\t",b[i]);
}
return 0;
}
```

Finding NCR and NPR. **(2018-08-16 11:57)**

```
long find _ncr(int , int)

long find _npr(int , int)
long factorial(int)
int main()
{
int n,r;
long ncr,npr;
printf("Enter the value of n and r\n");
scanf(" %d %d",n,r);
ncr = find _ncr(n,r);
npr = find _npr(n,r);
printf(" %dC %d= %ld\n",n,r,ncr);
printf(" %dC %d= %ld\n",n,r,npr);
return 0;
}
long find _ncr(int n, int r)
{
long result;
```

```
result = factorial(n)/factorial(n-r)*factorial(r);
return result;
}
long find _npr(int n, int r)
{
long result;
result = factorial(n)/factorial(n-r);
return result;
}
long factorial(int n) {
if(n==1||n==0)
return 1;
else
return n*factorial(n-1);
}
```



Structures in C Language **(2018-08-16 15:33)**

C Programming Structure

In this article you will learn about structures in C language.What is it, how to define it and use it in your program.

structure: It is a Collection of variables of different types under a single name.

```
struct student {
char name[50];
int age;
float percentage;
};
```

Here *Struct* is the keyword used to define structure and student is the name of the structure.

Creating Structure Variable:

```
struct student {
char name[50];
int age;
float percentage;
}struct student s1,s2,s3;
```

Here S1,S2,S3 are the structure variables.We can access the members of the structures using these variables.

There are two types of operators used for accessing members of a structure.

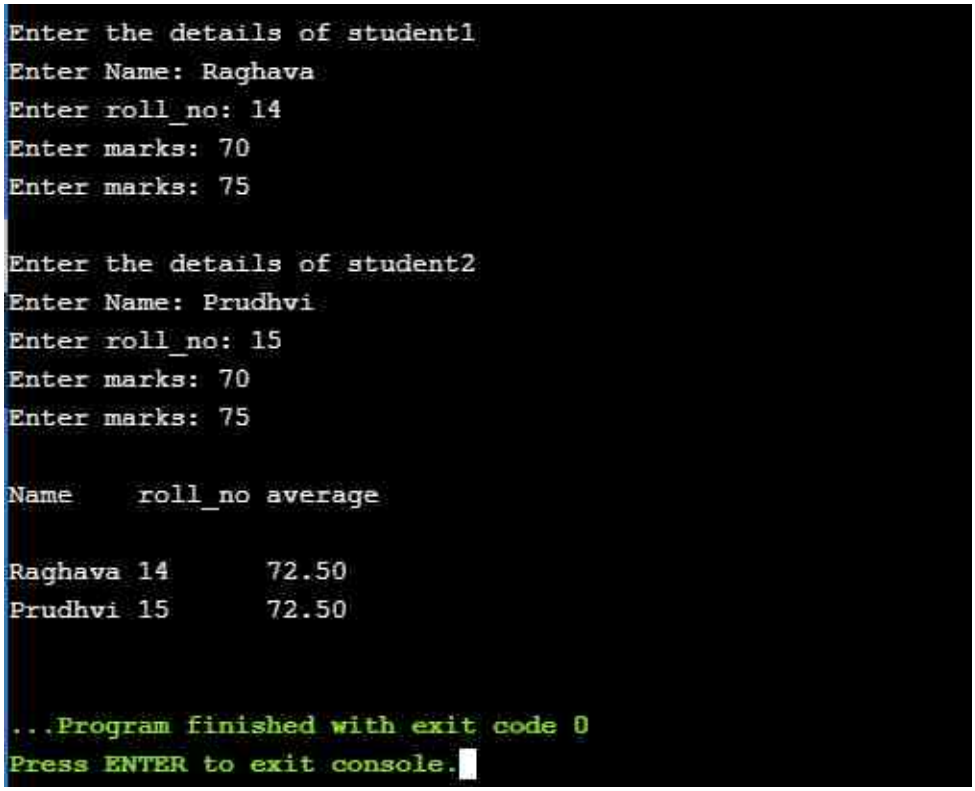
Member operator(.)

Structure pointer operator(->)

program representing array of structures

```
#include<stdio.h>
#include<string.h>
#define MAX 2
#define SUBJECTS 2
struct student {
char name[20];
int roll _no;
float marks[SUBJECTS];
};
int main()
{
struct student arr _student[MAX];
int i,j;
float sum = 0;
for(i=0;i<MAX;i++)
{
printf("\nEnter the details of student %d\n",i+1);
printf("Enter Name: ");
scanf(" %d", &arr _student[i].name);
printf("Enter roll _no: ");
scanf(" %d", &arr _student[i].roll _no);
for(j=0;j<SUBJECTS;j++)
{
printf("Enter marks: ");
scanf(" %d", &arr _student[i].marks[j]);
}
```

```
}
printf("\n");
printf("Name\troll _no\taverage\n\n");
for(i = 0; i < MAX; i++ )
{
sum = 0;
for(j = 0; j < SUBJECTS; j++)
{
sum = sum+ arr _student[i].marks[j];
}
printf(" %s\t %d\t %.2f\n",arr _student[i].name, arr _student[i].roll _no, sum/SUBJECTS);
}
return 0;
}
```



1.2 September

Median (2018-09-09 22:57)

/*Median of given numbers*/
#include <stdio.h>

```
int main() {
int n=4,k,a[10]= {1,2,3,4,5 };
k=n/2;
if(n %2!=0)
{
printf(" %d",a[k+1]);
}
else if(n %2==0)
{
int x= a[k]+a[(k/2)+1];
float y = x/2;
printf(" %.2f",y);
}
return 0;
}
```

Result

```
$gcc -o main *.c
$main
3.00
```

1.3 October

Printing prime Number between a range 2 to 10 (2018-10-01 10:17)

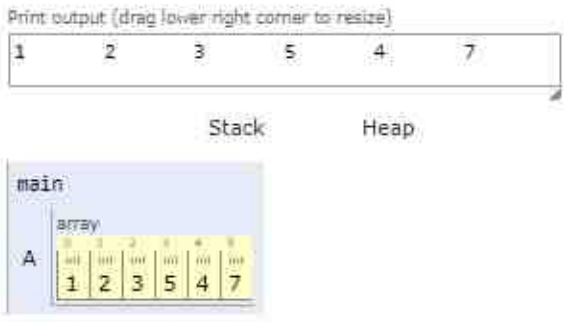
```
int main() {
int i,j;
for(i=2;i<=10;i++)
{
for(j=2;j<=i;j++)
{
if(i %j==0)
{
break;
}
}
if(i==j)
{
printf(" %d\t",j);
}
}

return 0;
```

```
}
```

OUTPUT:

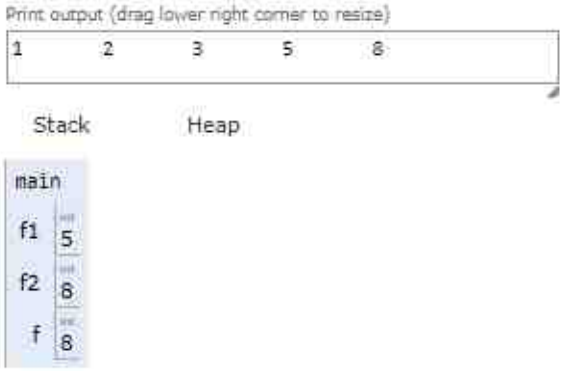
Selection sort **(2018-10-02 09:47)**

```
void selectionSort(int A[],int n)
{
for(int i=0;i<=4;i++)
{
int imin=i;
for(int j=i+1;j<=5;j++)
{
if(A[j]<A[imin])
{
imin = j;
}
int temp = A[i];
A[i] = A[imin];
A[imin] = temp ;
}
}
}
int main() {
int A[]= {2,4,7,1,5,3 };
selectionSort(A,6);
for(int i=0;i<=5;i++)
{
printf(" %d",A[i]);
}
return 0;
}
```



Fibbonacci series (2018-10-02 11:38)

```
int main() {
int f1 =0,f2=1,f;
for(int i=0;i<5;i++)
{
f = f1 +f2;
f1=f2;
f2=f;
printf(" %d\t",f);
}
return 0;
}
```



Converting Celsius to Fahrenheit and Fahrenheit to Celsius (2018-10-04 14:15)

```
#include <stdio.h>
int main() {
```

```
float c,f;
int x;
//press 1 if u want temperature in celsius
//press 2 if u want temperature in fahrenheit
printf("Enter your choice 1 || 2 \n");
scanf(" %d", &x);
switch(x)
{
case 1:
printf("Enter the value of temperature in farenheit\n");
scanf(" %f", &f);
c=f-32;
printf("The value of temperature in celsius is %1.f",c/9);
break;
case 2:
printf("enter the value of temperature in celsius\n");
scanf(" %f", &c);
f= 9*c;
printf("The value of temperature in fahrenheit is %1.f",f+32);
break;
return 0;
}
}
```

HANGMAN GAME IN JAVASCRIPT (2018-10-10 11:44)

```
<html>
<script>
alert("Note: This ia a Game where You will guess words by using input as letters you shoud give input based on blankspaces & & your guessing is limited to 15");
var words= ["javascript","monkey","amazing","pancake","album",
"atlas",
"bestseller",
"booklet",
"brochure",
"codex",
"compendium",
"copy",
"dictionary",
"dissertation",
"edition",
"encyclopedia",
"essay",
"fiction",
"folio",
"handbook",
"hardcover",
"leaflet",
```

```
"lexicon",
"magazine",
"manual",
"monograph","nonfiction",
"novel",
"octavo",
"offprint",
"omnibus",
"opus",
"opuscule",
"pamphlet",
"paperback",
"periodical",
"portfolio",
"preprint",
"primer",
"publication",
"quarto",
"reader",
"reprint",
"roll",
"scroll",
"softcover",
"speller",
"text",
"textbook",
"thesaurus",
"tome",
"tract",
"treatise",
"vade mecum",
"volume",
"work",
"writing"];
var word= words[Math.floor(Math.random()*words.len gth)];
var answerArray= [];
for(var i=0;i< word.length;i++)
{
answerArray[i]= " _ ";
}


var remainingLetters= word.length;
var gusses=0
alert("Blank spaces will be displayed i.e, a n no.of letter word u need to guess");
while(remainingLetters>0 & &gusses<15)
{



alert(answerArray.join(" "));
```

```
var guess= prompt("Guess a letter,or Click Cancel to stop playing.If ur guess is incoorect blank spaces will be displayed");

gusses++;



if(guess===null)
{

break;

}

else if(guess.length!==1)
{

alert("Please enter a single letter");
}

else

{
for(var j=0;j<word.length;j++)
{

if(word[j]===guess.toLowerCase())
{

answerArray[j] = guess.toLowerCase();

if(answerArray[j]===" _")
{

remainingLetters–;
}
}
}
}
```

```
}

alert(answerArray.join(" "));

alert("Good job! The answer was " + word);
</script>
</html>
```

Print 2-D array in spiral order **(2018-10-30 12:04)**

```
#include <stdio.h>
int main() {
int arr[3][3]= {
{1, 2, 3 },
{4, 5, 6 },
{7, 8, 9 },
};
spiral(arr);
return 0;
}
void spiral(int arr[3][3])
{
int top =0,right=2,left=0,bottom=2,dir =1, i;

while(top <= bottom & & left <= right)
{
if (dir==1)
{
for( i =0;i<=2;++i)
{
printf(" %d",arr[top][i]);
}
++top;
dir=2;
}

else if(dir==2)
{
for(i = top;i<=2;++i)
{
printf(" %d",arr[i][right]);
}
–right;
dir=3;
}
```

```
else if(dir==3)
{
for(i=top;i>=left;–i)
{
printf(" %d",arr[bottom][i]);
}
–bottom;
dir=4;
}

else if(dir==4)
{
for(i=bottom;i>=top;–i)
{
printf(" %d",arr[i][left]);
}
++left;
dir =1;
}

}
}
```

Expected Output:
123698745

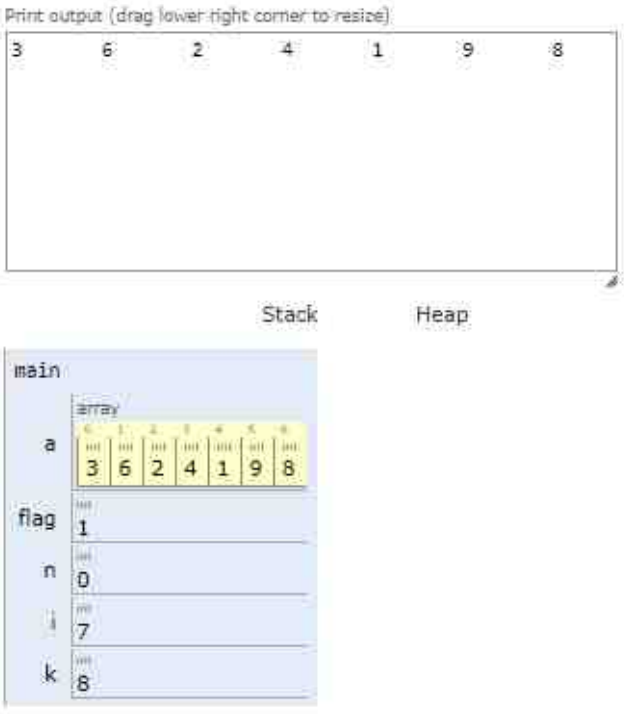
1.4 November

print an array in zigzag order (2018-11-01 11:23)

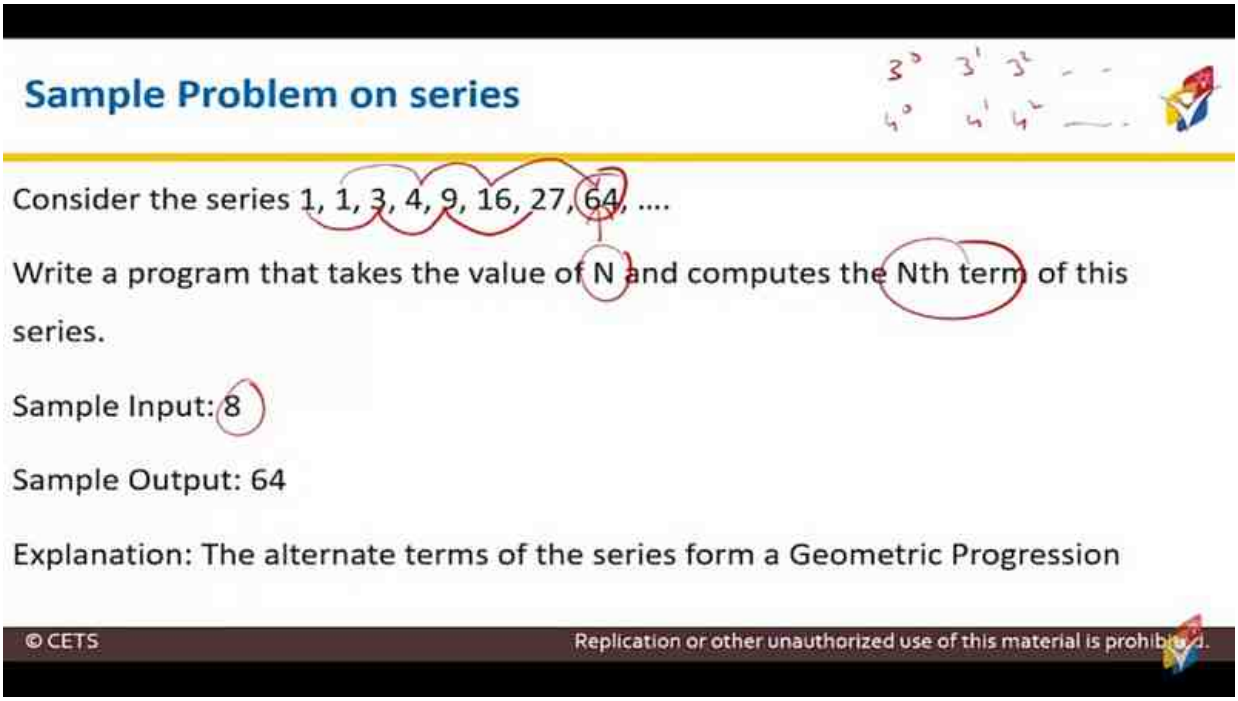
```
int main() {
int a[7]= {3,4,6,2,1,8,9 };
int flag =0,n=3,i=0,k;
while(n)
{
if(flag==0 & &a[i]<a[i+1])
{

flag=1;
i++;
}
else {
k=a[i];
a[i]=a[i+1];
```

```
a[i+1]=k;
i++;

}
if(flag==1 & &a[i]>a[i+1])
{
flag=0;
i++;
}
else
{
k=a[i];
a[i]=a[i+1];
a[i+1]=k;
i++;

}

n−;
}
for(i=0;i<7;i++)
{
printf(" %d\t",a[i]);
}

}
```

OUTPUT:



Geometric progression (2018-11-06 14:33)




```
#include <stdio.h>
int main() {
long int a,r;
printf("Enter a\n");
scanf(" %ld", &a);
printf("Enter r\n");
scanf(" %ld", &r);
long int k=1,n;
printf("Enter n\n");
scanf(" %ld", &n);
if(r==3||r==4)
{
if(n %2!=0)
{
n = (n/2)+1;
for(long i=1;i<n;i++)
{
k=k*r;
}
k= a*k;
printf(" %d",k);
```

```
}
else
{
n = n/2;
for(long i=1;i<n;i++)
{
k=k*r;
}
k= a*k;
printf(" %d",k);
}
}
}
```

OUTPUT:

Enter a

1

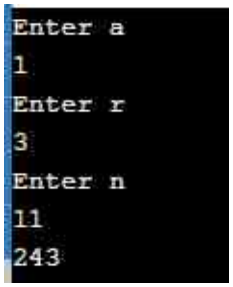
Enter r

3

Enter n

11

243



series of numbers when a sequence of prime numbers are added (2018-11-09 11:54)

The program takes two inputs – the value of the first term and the value of N and gives the Nth term of the series as the output.

Sample Input: 4 7

Sample Output:

Explanation:

Second Term = 4+2=6; Third Term = 6+3=9; Fourth Term= 9+5=14; Fifth Term=14+7=21; Sixth Term= 21+11=32; Seventh Term= 32+13=45

```
int main() {
int i,j,k=4;
for(i=2;i<=13;i++)
{
for(j=2;j<=i;j++)
{
if(i %j==0)
{
break;
}
}
if(i==j)
{
k= k+i;
printf(" %d ",k);
}
}
return 0;
}
```

OUTPUT:

Accept a positive integer and find next prime number to it (2018-11-09 12:33)

Write a program to accept a positive integer and find the immediate next prime number after it.

Sample Input:

8

Sample Output:

11

```
#include <stdio.h>
int main()
{
int i,j,x;
printf("Enter The Value of X");
scanf(" %d", &x);
for(i=x+1;i<=x+10;i++)
{
for(j=2;j<=i;j++)
{
if(i %j==0)
{
break;
}
}
if(i==j)
{
printf(" %d",i);
break;
}
}
```

OUTPUT:

Enter The Value of X
8
11

```
Enter The Value of X
8
11
```

---

write a program that takes the first term and the value n and kth term in the series (2018-11-09 13:59)

A series formed such that each term is formed by taking the previous term and adding 5 or subtracting 2 alternately.

For example, the series can be 4, 9, 7, 12, 10, 15, 13 ....

Write a program that takes the first term and the value of N and computes the nth term of this series.

Input: 4 5

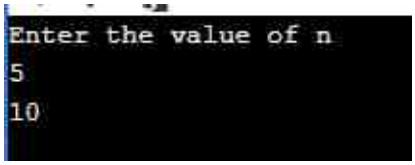
Ouptut: 10

```
#include <stdio.h>
int main()
{
int i,n,k=4;
printf("Enter the value of n");
scanf(" %d", &n);
for(i=2;i<=n;i++)
{
if(i %2==0)
{
k=k+5;
}
else
k=k-2;
}
printf(" %d",k);
```

```
return 0;
}
```

OUTPUT:

```
Enter the value of n
5
10
```



2. 2019

2.1 January

Program to know the range of signed integer (2019-01-14 16:55)

```
/* Program to know the range of signed integer*/
#include <stdio.h>
#include <limits.h>
int main() {
int var1 = INT _MIN;
int var2 = INT _MAX;
printf("range of signed integer is from %d to %d",var1,var2);
return 0;
}
```

Program to know the range of unsigned integer (2019-01-14 16:57)

```
/* Program to know the range of unsigned integer*/
#include <stdio.h>
#include <limits.h>
int main() {
int var1 = 0;
int var2 = UINT _MAX;
printf("range of unsigned integer is from %u to %u",var1,var2);
return 0;
}
```

Program to know the range of short signed integer (2019-01-14 17:02)

```
/* Program to know the range of short signed integer*/
#include <stdio.h>
#include <limits.h>
int main() {
short int var1 = SHRT _MIN;
short int var2 = SHRT _MAX;
printf("range of short signed integer is from %d to %d",var1,var2);
return 0;
}
```

Program to know the range of short unsigned integer **(2019-01-14 17:04)**

```
/* Program to know the range of short unsigned integer*/
#include <stdio.h>
#include <limits.h>
int main() {
short int var1 = 0;
short int var2 = USHRT _MAX;
printf("range of short unsigned integer is from %u to %u",var1,var2);
return 0;
}
```

Adding two numbers without using plus(+) operator **(2019-01-15 13:26)**

```
int main() {
int x = 3 , y = 4;
while(y) {
x++;
y−;
}
printf(" %d",x);

return 0;
}
```

output: 7

Adding Negative numbers without using plus operator **(2019-01-15 14:33)**

```
int main() {
int x = -1, y = -4;
while(y!=0)
{
x−;
y++;

}
printf(" %d",x);

return 0;
}
```

output: -5

Binary search **(2019-01-17 04:59)**

```
int main() {
int A[] = {2,4,6,8,10,11 };
int x = 4;
int n = 6;
int index = binarySearch(A,n,x);
if(index == -1)
{
printf(" %d is not found at index %d",x,index);
}
else {
printf(" %d is found at index %d",x,index);
}
return 0;
}
int binarySearch(int A[],int n,int x) {
int start = 0, end = n-1;
while(start <= end) {
int mid = (start + end)/2;
if(x == A[mid]) {
return mid;
}
else if(x < A[mid]) {
end = mid-1;
}
else {
start = mid+1;
}
}
return -1;
}
```

Finding first or last occurrence of an element in array **(2019-01-17 06:06)**

```
#include <stdio.h>

int main() {
int A[] = {2,2,6,8,10,11 };
int x = 2;
int n = 6;
int result = -1;
int index = binarySearch(A,n,x);
if(index == -1)
{
printf(" %d is not found at index %d",x,index);
}
else {
```

```
printf(" %d is found at index %d",x,index);
}
return 0;
}
int binarySearch(int A[],int n,int x) {
int start = 0, end = n-1, option,result = -1;
printf("Choose option:1 for first occurrence \nChoose option:2 for last occurrence\n");
printf("Enter any option to find first or last occurrence of an element:");
scanf(" %d", &option);
while(start <= end) {
int mid = (start + end)/2;

if(option == 1) {
if(x == A[mid]) {
result = mid;
end = mid-1;
}
else if(x < A[mid]) {
end = mid-1;
}
else {
start = mid+1;
}
}

if(option == 2) {
if(x == A[mid]) {
result = mid;
start = mid+1;
}
else if(x < A[mid]) {
end = mid-1;
}
else {
start = mid+1;
}
}
}
return result;
}
```

OUTPUT:

```
Choose option:1 for first occurrence
Choose option:2 for last occurrence
Enter any option to find first or last occurrence of an element:1
2 is found at index 0
```

search element in a circular sorted array **(2019-01-17 08:32)**

```
int main() {
int A[] = {12,14,18,21,3,6,8,9 };
int n = 8;
int x = 6;
int index = binarySearch(A,n,x);
if(index == -1) {
printf(" %d not found in the array",x);
}
else {
printf(" %d is found at index %d",x,index);
}
return 0;
}
int binarySearch(int A[],int n, int x) {
int low = 0,high = n-1;
while(low <= high) {
int mid = (low + high)/2;
if(x == A[mid])
return mid;

if(A[mid]<=A[high]) {
if(x>A[mid] & & x<=A[high])
low = mid+1;
else
high = mid-1; }
else {
if(x>=A[low] & & x <A[mid])
high = mid-1;
else
low = mid+1; }
}
return -1;
}
```

BlogBook v1.1,
LaTeX 2$_\varepsilon$ & GNU/Linux.
https://www.blogbooker.com

Edited: February 4, 2019