Array Practice Questions

1. WAP to input an array of N number of elements and display it.

#include <stdio.h> int main()

int i,n;

printf(”Enter size of array\n”); scanf(“%d“,&n);

int a[n];

printf(“Enter %d elements of array:\n”,n); for(i=0;i<=n-1;i++)

scanf(”%d“,&a[i]);

printf(”Entered Array is;\n“); for(i=0;i<=n-1;i++)

printf(“%d\t“,a[i]);

1. WAP to input an array of N number of elements and display it in reverse order.

#include <stdio.h> int main()

int i,n;

printf(”Enter size of array\n“); scanf(”%d“,&n);

int a[n];

printf(”Enter %d elements of array:\n”,n); for(i=0;i<=n-1;i++)

scanf(”%d“,&a[i]);

printf(”Entered Array is;\n“); for(i=n-1;i>=0;i--)

printf(“%d\t“,a[i]);

## WAP to input an array of N number of elements and find the sum and average of all the elements of that array.

#include <stdio.h>

int main()

int i,s=0,n; float avg;

printf(“Enter size of array\n“); scanf(”%d“,&n);

int a[n];

printf(“Enter %d elements of array:\n“,n); for(i=0;i<=n-1;i++)

scanf(”%d“,&a[i]); s=s+a[i];

avg=(float)s/n; printf(“Average = %f\n“,avg);

1. WAP to input an array of N number of elements and count total number of positives, negatives and zero elements in that array and display those counts.

#include <stdio.h>

int main()

int i,pos=0,neg=0,zero=0,n; float avg;

printf(”Enter size of array\n“); scanf(“%d“,&n);

int a[n];

printf(”Enter %d elements of array:\n”,n); for(i=0;i<=n-1;i++)

scanf(”%d“,&a[i]); if(a[i]>0)

pos++;

else if(a[i]<0) neg++;

else zero++;

printf(”Postive = %d\nNegative = %d\nZeros = %d\n”,pos,neg,zero);

1. WAP to input an array of N number of elements and store all even numbers in 1 array and all odd numbers in another array. Print both the even and odd array separately.

#include <stdio.h>

int main()

int i,n,c=-1,d=-1;

printf(”Enter size of array\n“); scanf(”%d“,&n);

int a[n],even[n],odd[n];

printf(”Enter %d elements of array:\n”,n); for(i=0;i<=n-1;i++)

scanf(”%d“,&a[i]); if(a[i]%2==0)

c++;

even[c]=a[i];

else

d++;

odd[d]=a[i];

printf(”Even Array:\n”); for(i=0;i<=c;i++)

printf(“%d\t“,even[i]); printf(“\nOdd Array:\n“); for(i=0;i<=d;i++)

printf(“%d\t“,odd[i]);

1. WAP to input an array of N number of elements and find their standard deviation.

#include <stdio.h>

#include <math.h> int main()

int i,s=0,n,r=0;

## float avg,sd;

printf(”Enter size of array\n“); scanf(”%d”,&n);

int a[n];

printf(”Enter %d elements of array:\n”,n); for(i=0;i<=n-1;i++)

scanf(”%d“,&a[i]); s=s+a[i];

avg=(float)s/n; for(i=0;i<=n-1;i++)

r=r+pow(avg-a[i],2); sd=sqrt(r/n);

printf(”Standard Deviation = %f\n”,sd);

1. Suppose there is president election in US and there are 2 candidates Trump and Biden. Input the votes of both the candidates in 10 states of US and calculate state-wise winner and overall winner.

#include <stdio.h> int main()

int i,a[10],b[10],c=0,d=0;

printf(”Enter votes of candidate A and B respectively\n”); for(i=0;i<=9;i++)

printf(“Enter votes of state no. %d\n”,i+1); scanf(“%d“,&a[i]);

scanf(“%d“,&b[i]);

if(a[i]>b[i])

c++;

printf(“Candidate A is winner in state no. %d\n”,i+1); else if(a[i]<b[i])

d++;

printf(“Candidate B is winner in state no. %d\n”,i+1);

else

printf(“Tie in state no. %d\n”,i+1);

c++, d++;

if(c>d)

printf(“Overall Winner is Candidate A\n”); else if(c<d)

printf(“Overall Winner is Candidate B\n”); else

printf(“Overall Tie\n“);

1. WAP to read the marks of 500 students of a course in computer programming and print the frequency of each score above 60. Do it using most efficient method you could taking minimum memory and minimum time.

#include <stdio.h>

int main()

int i,n,marks;

printf(“Enter number of students\n”); scanf(”%d“,&n);

int a[40]={0};

printf(“Enter marks of %d students out of 100:\n“,n); for(i=0;i<=n-1;i++)

scanf(“%d“,&marks); if(marks>60)

a[marks-61]++;

for(i=0;i<=40-1;i++)

if(a[i]>0)

printf(“Frequency of %d score = %d\n”,i+61,a[i]);

1. WAP to input an array of N number of elements and find the largest element in that array.

#include <stdio.h>

int main()

int i,n,max;

printf(”Enter number of elements\n”); scanf(”%d“,&n);

int a[n]; for(i=0;i<=n-1;i++)

scanf(“%d“,&a[i]);

max=a[0]; for(i=0;i<=n-1;i++)

if(a[i]>max)

max=a[i];

printf(”Largest Element = %d\n”,max);

# WAP to input an array of N number of elements and find the smallest element in that array.

#include <stdio.h> int main()

int i,n,min;

printf(”Enter number of elements\n”); scanf(”%d“,&n);

int a[n]; for(i=0;i<=n-1;i++)

scanf(“%d“,&a[i]);

min=a[0]; for(i=0;i<=n-1;i++)

if(a[i]<min)

min=a[i];

printf(”Smallest Element = %d\n”,min);

# WAP to input an array of N number of elements and swap the largest and smallest element in that array and print the updated array.

#include <stdio.h>

int main()

int i,n,min,max,c=0,d=0,t; printf(”Enter number of elements\n”); scanf(“%d“,&n);

int a[n]; for(i=0;i<=n-1;i++)

scanf(“%d“,&a[i]);

min=a[0],max=a[0]; for(i=0;i<=n-1;i++)

if(a[i]<min)

min=a[i];

C=l;

if(a[i]>max)

max=a[i];

d=1;

t=a[c];

a [c] =a[d] ;

a[d]=t;

printf(”Array after swaping smallest and largest element:\n”); for(i=0;i<=n-1;i++)

printf(“%d\t“,a[i]);

1. WAP to input an array of N number of elements and find the second smallest element and 2nd largest element in that array.

#include <stdio.h>

int main()

int i,n,min,smin,max,smax; printf(“Enter number of elements\t”); scanf(”%d“,&n);

int a[n];

for(i=0;i<=n-1;i++) scanf(“%d“,&a[i]);

if(n==1)

min=a[0]; smin=a[0]; max=a[0]; smax=a[0];

else if(a[0]<a[1])

min=a[0]; smin=a[1]; max=a[1]; smax=a[0];

else

min=a[1]; smin=a[0]; max=a[0]; smax=a[1];

for(i=2;i<=n-1;i++) if(a[i]<min)

m n m n

l9 a

else if (a[i]<smin) smin=a[i];

if(a[i]>max)

smax=max; max=a[i];

else if (a[i]>smax) smax=a[i];

printf(”Second Smallest No. = %d\n”,smin);

printf(”Second Largest No. = %d\n”,smax);

1. WAP to input an array of N number of distinct elements. Input an element you want to search and find it. If found then print the position of that element otherwise print not found.

#include <stdio.h>

int main()

int i,n,ele;

printf(“Enter number of elements\n”); scanf(”%d“,&n);

int a[n]; for(i=0;i<=n-1;i++)

scanf(“%d“,&a[i]);

printf(“Enter element to search\n”); scanf(”%d“,&ele);

for(i=0;i<=n-1;i++) if(a[i]==ele)

printf(“Element found at position %d\n”,i+1); break;

if(i==n)

printf(”Element not found\n”);

1. WAP to input an array of N number of elements (Elements can repeat) . Input an element you want to search and find it. If found then print all the positions of that element otherwise print not found.

#include <stdio.h>

int main()

int i,n,ele,c=0;

printf(”Enter number of elements\n”); scanf(”%d“,&n);

int a[n]; for(i=0;i<=n-1;i++)

scanf(“%d“,&a[i]);

printf(”Enter element to search\n”); scanf(”%d“,&ele);

for(i=0;i<=n-1;i++) if(a[i]==ele)

printf(“Element found at position %d\n”,i+1);

c =1;

if(i==0)

printf(“Element not found\n”);

# WAP to input an array of N number of elements and sort it in ascending order using bubble sort.

#include <stdio.h> int main()

int i,j,n,t;

printf(”Enter size of array\n“); scanf(“%d“,&n);

int a[n];

printf(”Enter %d elements of array:\n”,n); for(i=0;i<=n-1;i++)

scanf(”%d“,&a[i]);

for ( 1=1; 1‹=n -1; 1++)

for(j=0;j<=n-i-1;j++) if(a[j]>a[j+1])

t=a[j]; a[j]=a[j+1]; a[j+1]=t;

printf(”Sorted Array\n“); for(i=0;i<=n-1;i++)

printf(”%d\t“,a[i]);

# WAP to input an array of N number of elements and sort it in descending order using bubble sort.

#include <stdio.h> int main()

int i,j,n,t;

printf(”Enter size of array\n“); scanf(”%d“,&n);

int a[n];

printf(“Enter %d elements of array:\n”,n); for(i=0;i<=n-1;i++)

scanf(”%d“,&a[i]);

for ( 1=1; 1‹=n -1; 1++)

for(j=0;j<=n-i-1;j++) if(a[j]<a[j+1])

t=a[j]; a[j]=a[j+1]; a[j+1]=t;

printf(”Sorted Array\n“); for(i=0;i<=n-1;i++)

printf(“%d\t“,a[i]);

1. WAP to input an array of N number of elements. Input an element you want to insert in that array along with the position and insert it. Print the final array after insertion.

#include <stdio.h>

int main()

int i,n,ele,pos;

printf(”Enter size of array\n“); scanf(“%d“,&n);

int a[n+1];

printf(”Enter %d elements of array:\n”,n); for(i=0;i<=n-1;i++)

scanf(”%d“,&a[i]);

printf(”Enter an element and its position:\n”); scanf(”%d%d“,&ele,&pos); if(pos>=1&&pos<=(n+1))

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

a [ 1]=a[ 1-1] ;

a[pos-1]=ele;

n++;

else

printf(“Position Entered out of limit Hence cannot insert\n”);

for(i=0;i<=n-1;i++) printf(“%d\t“,a[i]);

1. WAP to input an array of N number of elements. Input E no. of elements you want to insert in that array along with their positions and insert all of them. Print the final array after insertion of all elements.

#include <stdio.h>

int main()

int i,j,n,ele,pos,E; printf(”Enter size of array\n“);

scanf(“%d“,&n); int a[n+1000];

printf(”Enter %d elements of array:\n”,n); for(i=0;i<=n-1;i++)

scanf(“%d“,&a[i]);

printf(”Enter number of elements to insert in an array upto 1000:\n”); scanf(“%d“,&E);

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

printf(“Enter an element and its position to insert:\n”); scanf(“%d%d“,&ele,&pos);

if(pos>=1&&pos<=(n+1)) for(j=n;j>=pos;j--)

a[j]=a[j-1];

a[pos-1]=ele; n++;

else

printf(“Position Entered out of limit Hence cannot insert\n”);

printf(”Array After Insertion:\n”); for(i=0;i<=n-1;i++)

printf(“%d\t“,a[i]);

# WAP to input an array of N number of elements. Input the position of element you want to delete. Print the element deleted and updated array after deletion of that element.

#include <stdio.h> int main()

int i,n,pos;

printf(”Enter size of array\n“); scanf(”%d“,&n);

int a[n];

printf(”Enter %d elements of array:\n”,n);

for(i=0;i<=n-1;i++) scanf(”%d“,&a[i]);

printf(”Enter position of an element to delete:\n”); scanf(“%d“,&pos);

if(pos>=1&&pos<=n)

printf(“Element Deleted = %d\n“,a[pos-1]); for(i=pos-1;i<=n-2;i++)

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

n--;

else

printf(“Position Entered out of limit Hence cannot insert\n”);

for(i=0;i<=n-1;i++) printf(”%d\t“,a[i]);

# WAP to input an array of N number of elements. Input the element you want to delete and delete the first occurrence of that element from that array. Print the updated array.

#include <stdio.h> int main()

int i,n,ele,c=-1;

printf(”Enter size of array\n“); scanf(”%d“,&n);

int a[n];

printf(”Enter %d elements of array:\n”,n); for(i=0;i<=n-1;i++)

scanf(”%d“,&a[i]);

printf(”Enter an element to delete its first occurence:\n”); scanf(”%d“,&ele);

for(i=0;i<=n-1;i++) if(a[i]==ele)

C=l;

break ;

if ( c ! = - 1)

printf(“Element Deleted = %d\n”,ele); for(i=c;i<=n-2;i++)

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

n--;

else

printf(“Element Not present in array Hence cannot delete\n“);

for(i=0;i<=n-1;i++) printf(”%d\t“,a[i]);

# WAP to input an array of N number of elements. Input the element you want to delete and delete all occurrence of that element from that array. Print the updated array.

#include <stdio.h> int main()

int i,j,n,ele,c=0;

printf(”Enter size of array\n“); scanf(”%d“,&n);

int a[n];

printf(”Enter %d elements of array:\n”,n); for(i=0;i<=n-1;i++)

scanf(”%d“,&a[i]);

printf(”Enter an element to delete :\n”); scanf(“%d“,&ele);

for(i=0;i<=n-1;i++) if(a[i]==ele)

for(j=i;j<=n-2;j++)

a[j]=a[j+1];

n--; c=1;

if(c==0)

printf(”Element Not present in array Hence cannot delete\n”); else

printf(“Element Deleted = %d\n”,ele); for(i=0;i<=n-1;i++)

printf(“%d\t“,a[i]);

# WAP to input an array of N number of elements. Left rotate this array by R number of rotations and print the final array.

Example:- Suppose array is 4 5 3 9 1

After left rotation by 1 it will be 5 3 9 1 4

#include <stdio.h> int main()

int i,r,j,n,first;

printf(“Enter size of array\n“); scanf(”%d“,&n);

int a[n];

printf(”Enter %d elements of array:\n”,n); for(i=0;i<=n-1;i++)

scanf(”%d“,&a[i]);

printf(”Enter number of rotations\n”); scanf(”%d“,&r);

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

first=a[0]; for(j=0;j<=n-2;j++)

a[j]=a[j+1];

a[n-1]=first;

printf(”Array after rotation\n”); for(i=0;i<=n-1;i++)

printf(“%d\t“,a[i]);

# WAP to input an array of N number of elements. Right rotate this array by R number of rotations and print the final array.

Example:- Suppose array is 4 5 3 9 1

After Right rotation by 1 it will be 1 4 5 3 9

#include <stdio.h> int main()

int i,r,j,n,last;

printf(“Enter size of array\n“); scanf(”%d“,&n);

int a[n];

printf(”Enter %d elements of array:\n”,n); for(i=0;i<=n-1;i++)

scanf(”%d“,&a[i]);

printf(”Enter number of rotations\n”); scanf(“%d“,&r);

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

last=a[n-1]; for(j=n-1;j>=1;j--)

a[j]=a[j-1]; a[0]=last;

printf(“Array after rotation\n“); for(i=0;i<=n-1;i++)

printf(”%d\t“,a[i]);

# WAP to input an array of N number of elements and find the frequency of an inputted element in that array.

#include <stdio.h> int main()

int i,n,ele,c=0;

printf(”Enter size of array\n“); scanf(“%d“,&n);

int a[n];

printf(”Enter %d elements of array:\n”,n); for(i=0;i<=n-1;i++)

scanf(”%d“,&a[i]);

printf(”Enter number to find its frequency in array\n”); scanf(”%d“,&ele);

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

if(a[i]==ele) c++;

printf(”Frequency of %d = %d\n”,ele,c);

# WAP to input an array of N number of elements and find the frequency of all elements in that array.

#include <stdio.h> int main()

int i,j,n,t;

printf(”Enter size of array\n“); scanf(”%d“,&n);

int a[n];

printf(”Enter %d elements of array:\n”,n); for(i=0;i<=n-1;i++)

scanf(”%d“,&a[i]); for(i=1;i<=n-1;i++)

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

if(a[j]>a[j+1])

t=a[j]; a[j]=a[j+1]; a[j+1]=t;

printf(”Sorted Array\n“); for(i=0;i<=n-1;i++)

printf(”%d\t“,a[i]); printf(”\n“);

int c=1; for(i=0;i<=n-2;i++)

if(a[i]==a[i+1])

c++;

if(i==n-2)

printf(“Frequency of %d = %d \n”,a[i],c);

else

if(i==n-2)

printf(“Frequency of %d = %d \n”,a[n-2],c); printf(“Frequency of %d = %d \n”,a[n-1],c);

else

printf(“Frequency of %d = %d \n“,a[i],c);

c=1;

1. WAP to input an array of N number of elements . Traverse this array from starting to end , if element found is prime then convert it into palindrome number next to it and

if its not prime(composite) then convert it into next Armstrong number. print the updated array.

#include <stdio.h>

#include <math.h> int main()

int p,r,rev=0,s=0,i,j,k,n,c=0,d=0; printf(“Enter size of array\n“); scanf(”%d“,&n);

int a[n];

printf(“Enter %d elements of array:\n“,n); for(i=0;i<=n-1;i++)

scanf(”%d“,&a[i]); for(i=0;i<=n-1;i++)

c=0;

for ( j =2; j‹a [ i] ; j++)

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

c++;

if(c==0&&a[i]!=1) //prime for(k=a[i]+1;;k++)

p=k,rev=0; while(p>0)

r=p%10;

rev=rev\*10+r; p=p/10;

if(rev==k)

a [ 1]=rev;

break;

else //not prime

for(k=a[i]+1;;k++)

d=0,p=k,s=0; while(p>0)

d++;

p=p/10;

p=k;

while(p>0)

r=p%10;

s=s+pow(r,d); p=p/10;

if(s==k)

a [1] =s;

break;

printf(”Array after replacing prime with palindrome and non prime with amrstrong\n”);

for(i=0;i<=n-1;i++) printf(”%d\t“,a[i]);

1. WAP to input an array of N elements and delete all the elements from that array which are perfect number.

#include <stdio.h>

int main()

int i,j,k,s,n,ele;

printf(”Enter size of array\n“); scanf(“%d“,&n);

int a[n];

printf(”Enter %d elements of array:\n”,n);

for(i=0;i<=n-1;i++) scanf(”%d“,&a[i]); for(i=0;i<=n-1;i++)

s=0;

for(j=1;j<a[i];j++) if(a[i]%j==0)



if(s==a[i])

for(k=i;k<=n-2;k++)

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

n--;

printf(”Array after deleting perfect numbers:\n”); for(i=0;i<=n-1;i++)

printf(”%d\t“,a[i]);

# WAP to input an array of N number of elements and delete all the duplicate elements from that array.

#include <stdio.h> int main()

int i,j,k,n;

printf(”Enter size of array\n“); scanf(“%d“,&n);

int a[n];

printf(”Enter %d elements of array:\n”,n); for(i=0;i<=n-1;i++)

scanf(”%d“,&a[i]);

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

if (a[i]==a[j])

for ( k=j; k‹=n -2; k++)

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

n--;

printf(“Array after deleting all duplicate numbers: \n“); for(i=0;i<=n-1;i++)

printf(”%d\t“,a[i]);

1. Consider a scenario where there are two classes A and B having 30 students each. A test was conducted for both the classes in a single room and student having same class roll number but from different classes was made to sit together and as a result they copied from each other and scored equal marks. Wap in ‘C’ to compute the marks of class B students based on the marks of class A students (using Arrays).

#include <stdio.h>

int main()

int i;

int a[30],b[30];

printf(”Enter marks of 30 students of Section A:\n”); for(i=0;i<=29;i++)

scanf(”%d“,&a[i]); b[i]=a[i];

printf(”Marks of 30 students of Section B:\n”); for(i=0;i<=29;i++)

printf(”%d\t“,b[i]);

1. Write a program in ‘C’ to store (in an array) and print the roll numbers of students beginning from m to n.

#include <stdio.h>

int main()

int i,m,n,students;

printf(”Enter First and Last roll no.:\n”); scanf(”%d%d“,&m,&n);

students=n-m+1; int a[students];

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

a[i]=m; m++;

printf(“Roll numbers are:\n“); for(i=0;i<=students-1;i++)

printf(“%d\t“,a[i]);

1. Find the output of following . a.) #include<stdio.h>

int main()( int arr[5]; arr[0] = 5;

arr[2] = -10;

arr[3/2] = 2; arr[3] = arr[0];

printf("%d %d %d %d", arr[0], arr[1], arr[2], arr[3]); return 0;

OUTPUT: 5 2 -10 5

b.) #include<stdio.h> int main()(

int arr[2] = (10, 20, 30, 40, 50};

printf("%d %d %d %d %d",arr[0],arr[1],arr[2],arr[3],arr[-1]); return 0;

OUTPUT: 10 20 gv gv gv c.) #include<stdio.h>

int main()(

int arr[10] = (1,2,3};

printf("%d %d %d %d",arr[0],arr[2],arr[4],arr[6]); return 0;

OUTPUT: 1 3 0 0

# Q32. Consider an array [8] containing following elements- 12, 5, 17, 87, 109, 43, 44, 47

# Show the process of bubble sort on this array.

# Q32. Assume an array consists of following elements- 19 45 67 78 89 56

Apply linear search algorithm to search the element 78 on this array. Q34. Consider an array ar[7] -

12 67 45 34 87 90 23

What will be the contents of this array after execution of following code? Ior(i=2; i<=5;i++)

ar[i]=ar[i+1];

OUTPUT: 12 67 34 87 90 23 23

Q35. Consider an array ar[7] -

12 67 45 34 87 90 23

What will be the contents of this array after execution of following code?

Ior(i=7; i>=3;i--)

ar[i-1]=ar[i];

OUTPUT: 12 67 gv gv gv gv gv

Q36. What will be the output of following program?

int main(void) ( int i;

int a[5]=(1,2,3,4,5};

for(i=5;i>=2;i--)

a[i-1]=a[i-2];

for(i=0;i<5;i++) printf("%d ",a[i]); return 0;}

OUTPUT: 1 1 2 3 4

Q37. What will be the output of following program?

int main(void) ( int i;

int a[5]=(1,2,3,4,5};

for(i=5;i>=2;i--)

a[i+1]=a[i-2];

lor(i=0;i<5;i++) printf("%d ",a[i]); return 0;}

OUTPUT: 1 2 3 1 2