* Problem 1

Students Age Display

**AIM :-** To write a ‘c’ Program to take input of Age of Students and display the Age using **Array.**

**Theory :-** The collection of multiple Data in a single variable is known as **Array**. For example it is written as a[100];, a[5]={1,2,3,4,5};, a[]={1,2,3,4,5}; .

**Algorithms :-** Step 1:- Start.

Step 2:- Declare integer variable i and age for array.

Step 3:- Check FOR condition i<?, if it is true go to step 4 otherwise go to step 6.

Step 4:- Read variable age by For loop.

Step 5:- Increment i+1 and go to step 3.

Step 6:- Check FOR condition i<?, if it is true go to step 7 otherwise go to step 9.

Step 7:- Print age by for loop.

Step 8:- Increment i+1 and go to step 6.

Step 9:- Stop.

**Program :-**

#include<stdio.h>

int main()

{

int age[50];

int i;

printf("Enter the age of Students\n");

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

{

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

}

printf("The Age of Students is ......\n");

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

{

printf("Student No.%d = %d\n",i,age[i]);

}

return 0;

}

**Output :-** Enter the age of Students

12,13,10,14,15,18,17,16,9,7

The Age of Students is ......

Student No.0 = 12

Student No.1 = 13

Student No.2 = 10

Student No.3 = 14

Student No.4 = 15

Student No.5 = 18

Student No.6 = 17

Student No.7 = 16

Student No.8 = 9

Student No.9 = 7

**Observation :-** After performing the experiment we observed that Values can be stored in a single variable using Array and we were able to take input of many values using looping statements. Thus, reduced the time to write the scanf statement and print the values in the same manner using loop. It took 0.14 sec Compilation Time.

* Problem 2

**Smallest Number**

**AIM :-** To write a ‘c’ Program to take input of Numbers and find out the Smallest Number in the Array**.**

**Theory :-** The collection of multiple Data in a single variable is known as **Array**. For example it is written as a[100];, a[5]={1,2,3,4,5};, a[]={1,2,3,4,5}; .

**Algorithms :-** Step 1:- Start

Step 2:- Declare ‘a’ variable for array and i,j,b

Step 3:- Check For condition i<?, if it is true go to step 4 otherwise go to step 6.

Step 4:- Read variable ‘a’ by loop.

Step 5:- Increment i+1 and go to step 3.

Step 6:- Check For condition i<?, if it is true go to step 7 otherwise go to step 8.

Step 7:- if(a[i]>a[j]) Swap the values

b=a[i];

a[i]=a[j];

a[j]=b;.

Step 8:- Increment i+1 and go to step 6.

Step 8:- Print the smallest number in Array.

Step 8:- Stop.

**Program :-**

#include<stdio.h>

int main()

{

int a[15];

int i,j,b;

printf("Enter the Elements in the Array ::\n");

for(i=0;i<7;i++)

{

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

}

for(i=0;i<7;i++)

{

for(j=i+1;j<7;j++)

{

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

{

b=a[i];

a[i]=a[j];

a[j]=b;

}

}

}

printf("\t%d is the Smallest in the Array\n",a[0]);

return 0;

}

**Output :- Enter the Elements in the Array ::**

**56,76,34,23,13,32,15**

**13 is the Smallest in the Array**

**Observation :-** After performing the experiment we observed that Values can be stored in a One dimensional Array and we were able to find out the smallest Number in the Array by swapping the values. The smallest number is located at the 0th location and after that print the 0th location. It took 0.16 sec Compilation Time.

* Problem 3

**Even and Odd Numbers**

**AIM :-** To write a ‘c’ Program to take input of Numbers and find out the Even and odd Number in the Array**.**

**Theory :-** The collection of multiple Data in a single variable is known as **Array**. For example it is written as a[100];, a[5]={1,2,3,4,5};, a[]={1,2,3,4,5}; .

**Algorithms :-** Step 1:- Start

Step 2:- Declare ‘a[15]’ variable for array and i.

Step 3:- Check For condition i<?, if it is true go to step 4 otherwise go to step 6.

Step 4:- Read variable ‘a’ by loop.

Step 5:- Increment to i+1and go to step 3.

Step 6:- Check For condition i<?, if it is true go to step 7 otherwise go to step 9.

Step 7:- if(a[i]%2==0), print Even Numbers.

Step 8:- Increment to i+1 and go to step 6.

Step 9:- Check For condition i<?, if it is true go to step 10 otherwise go to step 12.

Step 10:- if(a[i]%2!=0), print Odd Numbers.

Step 11:- Increment to i+1 and go to step 9.

Step 12:- Stop.

**Program :-**

**#include<stdio.h>**

**int main()**

**{**

**int a[15];**

**int i;**

**printf("Enter the Elements in the Array ::\n");**

**for(i=0;i<7;i++)**

**{**

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

**}**

**printf("\n\nEven Numbers\n");**

**for(i=0;i<7;i++)**

**{**

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

**{**

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

**}**

**}**

**printf("\n\nOdd Numbers\n");**

**for(i=0;i<7;i++)**

**{**

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

**{**

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

**}**

**}**

**return 0;**

**}**

**Output :- Enter the Elements in the Array ::**

**4**

**3**

**76**

**98**

**35**

**32**

**45**

**Even Numbers**

**4**

**76**

**98**

**32**

**Odd Numbers**

**3**

**35**

**45**

**Observation :-** After performing the experiment we observed that Values can be stored in a One dimensional Array and we were able to find out the Even and Odd Number in the Array by Dividing the number by 2 and then checking the remainder of the Division. If it is 0 ,the number is Even and if it is 1 ,the number is odd. It took 0.30 sec Compilation Time.

* Problem 4

**Sum of Elements**

**AIM :-** To write a ‘c’ Program to take input of Numbers and find out the Sum of Number in the Array**.**

**Theory :-** The collection of multiple Data in a single variable is known as **Array**. For example it is written as a[100];, a[5]={1,2,3,4,5};, a[]={1,2,3,4,5}; .

**Algorithms :-** Step 1:- Start

Step 2:- Declare ‘a[15]’ variable for array and i, b=0.

Step 3:- Check For condition i<?, if it is true go to step 4 otherwise go to step 6.

Step 4:- Read variable ‘a’ by loop.

Step 5:- Increment to i+1and go to step 3.

Step 6:- Check For condition i<?, if it is true go to step 7 otherwise go to step 9.

Step 7:- calculate sum by b+=a[i];.

Step 8:- Increment to i+1 and go to step 6.

Step 9:- Print the value of ‘b’ variable.

Step 10:- Stop.

**Program :-**

**#include<stdio.h>**

**int main()**

**{**

**int a[15];**

**int i, b=0;**

**printf("Enter the Elements in the Array ::\n");**

**for(i=0;i<7;i++)**

**{**

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

**}**

**for(i=0;i<7;i++)**

**{**

**b+=a[i];**

**}**

**printf("The Sum of Values in the Array is ....%d\n",b);**

**return 0;**

**}**

**Output :- Enter the Elements in the Array ::**

**56**

**45**

**32**

**53**

**12**

**75**

**89**

**The Sum of Values in the Array is ....362**

**Observation :-** After performing the experiment we observed that Values can be stored in a One dimensional Array and we were able to find out the sum of Number in the Array by adding each element in the array and stored in b variable. It took 0.20 sec Compilation Time.

* Problem 5

**Average of Numbers**

**AIM :-** To write a ‘c’ Program to take input of Numbers and find out the Average of Numbers in the Array**.**

**Theory :-** The collection of multiple Data in a single variable is known as **Array**. For example it is written as a[100];, a[5]={1,2,3,4,5};, a[]={1,2,3,4,5}; .

**Algorithms :-** Step 1:- Start

Step 2:- Declare ‘a[15]’ float variable for array, b=0 and integer variable i.

Step 3:- Check For condition i<?, if it is true go to step 4 otherwise go to step 6.

Step 4:- Read variable ‘a’ by loop 7 times.

Step 5:- Increment to i+1and go to step 3.

Step 6:- Check For condition i<?, if it is true go to step 7 otherwise go to step 9.

Step 7:- calculate sum by b+=a[i];.

Step 8:- Increment to i+1 and go to step 6.

Step 9:- Print the value of ‘b’ variable.

Step 10:-Calculate Average by b/=7;.

Step 11:- Print the value of ‘b’ variable.

Step 11:- Stop.

**Program :-**

**#include<stdio.h>**

**int main()**

**{**

**float a[15],b=0;**

**int i;**

**printf("Enter the Elements in the Array ::\n");**

**for(i=0;i<7;i++)**

**{**

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

**}**

**for(i=0;i<7;i++)**

**{**

**b+=a[i];**

**}**

**printf("Sum :: %f\n",b);**

**b/=7;**

**printf("Average :: %f",b);**

**return 0;**

**}**

**Output :-**

**Enter the Elements in the Array ::**

**1**

**23**

**45**

**6**

**54**

**34**

**12**

**Sum :: 175.000000**

**Average :: 25.000000**

**Observation :-** After performing the experiment we observed that Values can be stored in a One dimensional Array and we were able to find out the Average of Numbers in the Array by adding each element in the array and stored in b variable. Then divide the sum with the number of elements present in the Array. It took 0.36 sec Compilation Time.