# CA3001 – Programming and Data Structures using C

# Assignment-11

**A. Array Manipulations using Pointers**

**One Dimensional Array:**

1. **Find Mean, Median, Mode, Variance, Standard Deviation, and Range of 'n' elements in an array**

#include <stdio.h>

#include<math.h>

int main()

{

int n,range,arr[100],temp,sum=0;

float mean,meadian,variance,std;

int mode, maxcount = 0, i, j;

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

scanf("%d",&n);

printf("enter the array elements:\n");

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

scanf("%d",arr+i);

}

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

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

if(\*(arr+i)<\*(arr+j))

{

temp=\*(arr+i);

\*(arr+i)=\*(arr+j);

\*(arr+j)=temp;

}

}

}

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

sum+=\*(arr+i);

}

mean=(float)sum/n;

printf("the mean of the array is:%f\n",mean);

if(n%2!=0){

meadian=\*(arr+n/2+1);

}

else{

meadian=(float)\*(arr+n/2)+\*(arr+n/2+1)/2;

}

printf("the meadian of the array is:%f\n",meadian);

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

int count = 0;

for (j = 0; j < n;j++) {

if (\*(arr+j) ==\*(arr+i)){

count++;

}

if (count > maxcount) {

maxcount = count;

mode= \*(arr+i);

}

}

}

printf("the mode of the array is:%d\n",mode);

sum=0;

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

sum+=(\*(arr+i)-mean)\*(\*(arr+i)-mean);

}

variance=(float)sum/n;

printf("the variance of the array is:%f\n",variance);

std=sqrt(variance);

printf("the standard deviation of the array is:%f\n",std);

range=\*(arr+n-1)-\*(arr+0);

printf("the range of the array is:%d\n",range);

return 0;

}

Output-

enter the size of array:10

enter the array elements:

1 2 2 4 3 3 3 5 6 6

the mean of the array is:3.500000

the meadian of the array is:5.000000

the mode of the array is:3

the variance of the array is:2.400000

the standard deviation of the array is:1.549193

the range of the array is:5

**2. Sort the 'n' elements of an array in Descending order**

**#include <stdio.h>**

int main() {

int arr[10],n,i,j,temp;

int \*ptr;

printf("enter the number of elements you want to insert\n");

scanf("%d",&n);

printf("enter the array elements:\n");

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

{

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

}

ptr=&arr[0];

printf("the array in reverse order is:\n");

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

{

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

if(ptr[i]<ptr[j])

{

temp=ptr[i];

ptr[i]=ptr[j];

ptr[j]=temp;

}

}

}

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

{

printf("%d\t",\*(ptr+i));

}

return 0;

}

Output-

enter the number of elements you want to insert

5

enter the array elements:

3 6 1 2 4

the array in reverse order is:

6 4 3 2 1

**3. Find the second largest and smallest element in an array.**

#include <stdio.h>

int main() {

int arr[10],n,i,j,temp;

int \*ptr;

printf("enter the number of elements you want to insert\n");

scanf("%d",&n);

printf("enter the array elements:\n");

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

{

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

}

ptr=&arr[0];

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

{

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

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

{

temp=ptr[i];

ptr[i]=ptr[j];

ptr[j]=temp;

}

}

}

printf("the 2nd largest number in the array is: %d",\*(ptr+n-2));

printf("\n the smallest number in the array is: %d",\*(ptr));

return 0;

}

Output-

enter the number of elements you want to insert

5

enter the array elements:

1 9 5 10 2

the 2nd largest number in the array is: 9

the smallest number in the array is: 1

**Two Dimensional Array:**

**4. Print the leading diagonal, upper triangular and lower triangular elements of mxm array**

#include <stdio.h>

int main()

{

int a[3][3],i,j,s=0;

printf("Enter Elements for 3\*3 Matrix:\n\n");

for(i=0;i<=2;i++)

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

{

scanf("%d",\*(a+i)+j);

}

printf("\n3\*3 Matrix :\n\n");

for(i=0;i<=2;i++)

{

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

{

printf("%d ",\*(\*(a+i)+j));

}

printf("\n");

}

printf("\n");

printf("the diagonal elements:\n");

for(i=0;i<=2;i++)

{

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

{

if(i==j)

{

printf("%d\t",\*(\*(a+i)+j));

}

else

printf("0\t");

}

printf("\n");

}

printf("\n");

printf("the lower triangular elements:\n");

for(i=0;i<=2;i++)

{

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

{

if(i<j)

{

printf("0\t");

}

else

printf("%d\t",\*(\*(a+i)+j));

}

printf("\n");

}

printf("\n");

printf("the upper triangular elements:\n");

for(i=0;i<=2;i++)

{

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

{

if(i>j)

{

printf("0\t");

}

else

printf("%d\t",\*(\*(a+i)+j));

}

printf("\n");

}

return 0;

}

Output-

Enter Elements for 3\*3 Matrix:

1 2 3 4 5 6 7 8 9

3\*3 Matrix :

1 2 3

4 5 6

7 8 9

the diagonal elements:

1 0 0

0 5 0

0 0 9

the lower triangular elements:

1 0 0

4 5 0

7 8 9

the upper triangular elements:

1 2 3

0 5 6

0 0 9

**5. Find the maximum & minimum element in each row and each coloumn of mxm array**

#include<stdio.h>

int main()

{

int i ,j ,mat[3][3];

printf("Enter Elements for 3\*3 Matrix:\n\n");

for(i=0;i<=2;i++)

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

{

scanf("%d",\*(mat+i)+j);

}

printf("\n3\*3 Matrix :\n\n");

for(i=0;i<=2;i++)

{

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

{

printf("%d ",\*(\*(mat+i)+j));

}

printf("\n");

}

printf("minimum in each row:\n");

for (int i = 0; i < 3; i++) {

int minm = \*(\*(mat+i)+0);

for (int j = 1; j < 3; j++) {

if (\*(\*(mat+i)+j) < minm)

minm = \*(\*(mat+i)+j);

}

printf("%d\t" ,minm);

}

printf("\nmaximun in each row:\n");

for (int i = 0; i < 3; i++) {

int maxm = \*(\*(mat+i)+0);

for (int j = 1; j < 3; j++) {

if (\*(\*(mat+i)+j) >maxm)

maxm = \*(\*(mat+i)+j);

}

printf("%d\t" ,maxm);

}

printf("\nminimum in each column:\n");

for (int i = 0; i < 3; i++) {

int minm = \*(\*(mat+0)+i);

for (int j = 1; j < 3; j++) {

if (\*(\*(mat+i)+j) < minm)

minm = \*(\*(mat+i)+j);

}

printf("%d\t",minm);

}

printf("\nmaximum in each column:\n");

for (int i = 0; i < 3; i++) {

int maxm = \*(\*(mat+0)+i);

for (int j = 1; j < 3; j++) {

if (\*(\*(mat+i)+j)> maxm)

maxm = \*(\*(mat+i)+j);

}

printf("%d\t",maxm);

}

return 0;

}

Output-

Enter Elements for 3\*3 Matrix:

1 2 3 4 5 6 7 8 9

3\*3 Matrix :

1 2 3

4 5 6

7 8 9

minimum in each row:

1 4 7

maximun in each row:

3 6 9

minimum in each column:

1 2 3

maximum in each column:

7 8 9

**6. Perform matrix multiplication between two mxm array**

#include <stdio.h>

int main() {

int m1[10][10], m2[10][10], m1m2[10][10], r, c;

printf("Enter rows and column for mxm matrix: ");

scanf("%d %d", &r, &c);

printf("\nEnter elements for m1: \n");

for (int i = 0; i < r; ++i) {

for (int j = 0; j < c; ++j) {

printf("Enter a%d%d: ", i + 1, j + 1);

scanf("%d", \*(m1+i)+j);

}

}

printf("\nEnter elements for m2: \n");

for (int i = 0; i < r; ++i) {

for (int j = 0; j < c; ++j) {

printf("Enter a%d%d: ", i + 1, j + 1);

scanf("%d",\*(m2+i)+j);

}

}

for (int i = 0; i < r; ++i) {

for (int j = 0; j < c; ++j) {

\*(\*(m1m2+i)+j) = 0;

}

}

for (int i = 0; i < r; ++i) {

for (int j = 0; j < c; ++j) {

for (int k = 0; k < c; ++k) {

\*(\*(m1m2+i)+j) += \*(\*(m1+i)+k) \* \*(\*(m2+k)+j);

}

}

}

printf("\nOutput Matrix:\n");

for (int i = 0; i < r; ++i) {

for (int j = 0; j < c; ++j) {

printf("%d ", \*(\*(m1m2+i)+j));

if (j == c - 1)

printf("\n");

}

}

return 0;

}

Output-

Enter rows and column for mxm matrix: 2 2

Enter elements for m1:

Enter a11: 1

Enter a12: 2

Enter a21: 2

Enter a22: 1

Enter elements for m2:

Enter a11: 3

Enter a12: 4

Enter a21: 4

Enter a22: 3

Output Matrix:

11 10

10 11

**B. String Manipulations using Pointers**

**7. Write a C Program to convert**

**a. Upper case to Lower case**

#include<stdio.h>

int main()

{

char str[100];

char \*s;

int i=0;

printf("Enter any string : ");

scanf("%s",str);

s=str;

while(s[i]!='\0')

{

if(s[i]>='A' && s[i]<='Z'){

s[i]=s[i]+32;

}

++i;

}

printf("String after stringLwr : %s\n",s);

return 0;

}

Output-

Enter any string : SAGA

String after stringLwr : saga

**b. Lower case to Upper case**

#include<stdio.h>

int main()

{

char str[100];

char \*s;

int i=0;

printf("Enter any string : ");

scanf("%s",str);

s=str;

while(s[i]!='\0')

{

if(s[i]>='a' && s[i]<='z'){

s[i]=s[i]-32;

}

++i;

}

printf("String after stringUpr : %s\n",s);

return 0;

}

Output-

Enter any string : saga

String after stringUpr : SAGA

**8. Write a C Program to read 2 string constants into a and b. Compare whether they are equal or not. if not, join them together. Then copy the contents of a to the variable c. At the end of the program, print the contents of all three variables and their length. (With and Without String Handling Functions).**

**A) with string handling.**

#include <stdio.h>

#include <string.h>

int main()

{ int result;

char a[100], b[100],c[100];

char \*str1,\*str2;

printf("Enter 'a' string:");

scanf("%s", a);

printf("Enter 'b' string:");

scanf("%s", b);

str1=a;

str2=b;

result=strcmp(str1,str2);

if(result!=0)

{

printf("%s and %s are not equal",str1,str2);

strcat(str1,str2);

}

else{

printf("%s and %s are equal",str1,str2);

}

strcpy(c,str1);

printf("\n") ;

printf("\n'a' is: %s\n'b' is: %s\n'c' is: %s",str1,str2,c);

return 0;

}

**b)without string handling.**

#include <stdio.h>

#include <string.h>

int main()

{

char a[100], b[100],c[100];

char \*str1,\*str2;

int i=0,flag=0,length;

printf("Enter 'a' string:");

scanf("%s", a);

printf("Enter 'b' string:");

scanf("%s", b);

str1=a;

str2=b;

while(str1[i]!='\0'||str2[i]!='\0')

{ if(str1[i]!=str2[i]){

flag=1;

break;

}

i++;

}

if(flag==1)

{

printf("%s and %s are not equal",str1,str2);

length=strlen(str1);

i=0;

while(str2[i]!='\0')

{

str1[length-1]=str2[i];

length++;

i++;

}

str1[length]='\0';

}

else{

printf("%s and %s are equal",str1,str2);

}

i=0;

while(str1[i]!='\0')

{

c[i]=str1[i];

i++;

}

c[i]='\0';

printf("\n");

printf("'a' is: %s\n'b' is: %s\n'c' is: %s",str1,str2,c);

return 0;

}

Output-

Enter 'a' string:saga

Enter 'b' string:rik

saga and rik are not equal

'a' is: sagrik

'b' is: rik

'c' is: sagrik

**9. Write a C program to read a string and prints if it is a palindrome or not.**

#include <stdio.h>

#include <string.h>

int main()

{

char string[20];

char \*str;

int i, length;

int flag = 0;

printf("Enter a string:");

scanf("%s", string);

length = strlen(string);

str=string;

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

{

if(str[i] != str[length-i-1])

{

flag = 1;

break;

}

}

printf("\n");

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

{

printf("%c",\*str);

str++;

}

if (flag) {

printf(" is not a palindrome");

}

else {

printf(" is a palindrome");

}

}

Output-

Enter a string:iillllii

iillllii is a palindrome

**C. Functions using Pointers**

**Write a C Program (Using Call by Value, Call by Reference & Category of Functions)**

**10. Check Prime and Armstrong Number by making function**

#include <stdio.h>

void prime(int \*);

int arms(int \*);

int main()

{

int n;

int result;

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

scanf("%d",&n);

prime(&n);

result=arms(&n);

if(result==1)

printf(" armstrong number");

else

printf(" not armstrong number");

return 0;

}

void prime(int \*n) {

int p,i=2;

do{

if(\*n%i==0||\*n==1)

{

p=0;

break;

}

i++;

}while(i<\*n/2);

if(p==0&&\*n!=2)

printf("%d is not a prime number\n",\*n);

else

printf("%d is a prime number\n",\*n);

}

int arms(int \*num){

int n,sum=0;

printf("%d number is",\*num);

while(\*num!=0){

n=\*num % 10;

\*num=\*num/10;

sum=sum+(n\*n\*n);

}

if(sum==\*num){

return 0;

}

else{

return 1;

}

}

Output-

enter a number:

66

is not a prime number

66 number is armstrong not number

11. Reverse a sentence using String Functions.

#include <stdio.h>

#include <string.h>

void reverse(char \*);

int main()

{

char string[20];

printf("Enter a string:");

scanf("%s", string);

reverse(string);

return 0;

}

void reverse(char \*str)

{

strrev(str);

printf("the reversed string is:%s",str);

}

Output-

Enter a string:saga

the reversed string is:agas