Bing

/Write a program in C to input a 2D array and represent it in Row and Column Major order/ #include<stdio.h> #define ROW 3 #define COL 3 int main() { int

**1./Write a program in C to input a 2D array and represent it in Row and Column Major order/**

#include<stdio.h>

#define ROW 3

#define COL 3

int main()

{

int arr[ROW][COL],i,j;

printf("Enter the elements in the 2D array\n");

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

{

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

{

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

}

}

printf("\nThe 2D array in ROW MAJOR order is\n");

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

{

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

{

printf("%d ",arr[i][j]);

}

printf("\n");

}

printf("\nThe 2D array in COLUMN MAJOR order is\n");

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

{

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

{

printf("%d ",arr[j][i]);

}

printf("\n");

}

return 0;

}

The output of this code will depend on the input that you give to the 2D array. For example, if you enter the following elements:

1 2 3 4 5 6 7 8 9

The output will be:

The 2D array in ROW MAJOR order is 1 2 3 4 5 6 7 8 9

The 2D array in COLUMN MAJOR order is 1 4 7 2 5 8 3 6 9

If you want to run the code and see the output yourself, you can use an online compiler like this one: <https://www.onlinegdb.com/online_c_compiler>

**2.//Write a program to insert an element in to the array**

#include<stdio.h>

void read\_array(int [], int); //array, size

void insert\_array(int [], int, int, int); //array, size, position, value

int main()

{

int a[30],n,value, pos;

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

scanf("%d",&n);

read\_array(a,n); //call by address

printf("Enter the position in which you want to insert a new value: ");

scanf("%d",&pos);

printf("Enter the new value: ");

scanf("%d",&value);

insert\_array(a,n,pos,value);

return 0;

}

void read\_array(int a[10], int n)

{

int i;

printf("Enter the numbers in to the array\n");

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

{

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

}

}

void insert\_array(int a[10], int n, int pos, int value)

{

int i;

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

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

a[pos]=value;

n=n+1;

printf("The array after inserting will be\n");

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

{

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

}

}

The output of this code will also depend on the input that you give to the array. For example, if you enter the following values:

Size of the array: 5 Numbers in the array: 10 20 30 40 50 Position to insert: 2 New value: 25

The output will be:

The array after inserting will be 10 25 20 30 40 50

Again, you can use an online compiler to run the code and see the output yourself.

**3.//Write a program to delete an element from the array**

#include<stdio.h>

void read\_array(int [], int);

void delete\_array(int [], int, int); //array, size, position

int main()

{

int a[30],n, pos;

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

scanf("%d",&n);

printf("In the main address of a is %x\n",a);

read\_array(a,n);

printf("Enter the position in which you want to delete: ");

scanf("%d",&pos);

delete\_array(a,n,pos);

return 0;

}

void read\_array(int p[30], int n)

{

int i;

printf("Enter the numbers in to the array\n");

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

{

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

}

printf("In the read\_array address of p is %x\n",p);

}

void delete\_array(int x[30], int n, int pos)

{

int i;

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

x[i]=x[i+1];

n=n-1;

printf("The array after deleting will be\n");

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

{

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

}

printf("In the delete\_array address of x is %x\n",x);

}

The output of this code will also depend on the input that you give to the array. For example, if you enter the following values:

Size of the array: 6 Numbers in the array: 11 22 33 44 55 66 Position to delete: 3

The output will be:

In the main address of a is 6c1e60 Enter the numbers in to the array In the read\_array address of p is 6c1e60 Enter the position in which you want to delete: The array after deleting will be 11 22 44 55 66 In the delete\_array address of x is 6c1e60

Note that the addresses of a, p and x may vary depending on your system. They are printed to show that they are pointing to the same memory location.

**4./C Program to add two polynomial using array/**

#include<stdio.h>

struct Polynomial

{

int coeff;

int exp;

};

struct Polynomial first[15], second[15], result[15];

void display(struct Polynomial poly[], int terms)

{

int i;

printf("\n");

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

{

printf("%dX^%d+ ", poly[i].coeff, poly[i].exp);

}

}

int readExpression(struct Polynomial poly[])

{

int terms, i;

printf("\nNumber of terms: ");

scanf("%d", &terms);

printf("\nEnter the coeffecients and exponents in DESCENDING order");

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

{

printf("\nCoeffecient :");

scanf("%d", &poly[i].coeff);

printf("Exponent :");

scanf("%d", &poly[i].exp);

}

return terms;

}

int addExpressions(int firstCount, int secondCount)

{

int i, j, k;

i = 0;

j = 0;

k = 0;

while(i < firstCount && j < secondCount)

{

if(first[i].exp == second[j].exp)

{

result[k].coeff = first[i].coeff + second[j].coeff;

result[k].exp = first[i].exp;

i++;

j++;

k++;

}

else if(first[i].exp > second[j].exp)

{

result[k].coeff = first[i].coeff;

result[k].exp = first[i].exp;

i++;

k++;

}

else

{

result[k].coeff = second[i].coeff;

result[k].exp = second[j].exp;

j++;

k++;

}

}

while(i < firstCount)

{

result[k].coeff = first[i].coeff;

result[k].exp = first[i].exp;

k++;

i++;

}

while(j < secondCount)

{

result[k].coeff = second[j].coeff;

result[k].exp = second[j].exp;

k++;

j++;

}

return k;

}

int main()

{

int firstCount, secondCount, resultCount;

printf("\nFirst Expression:\n");

firstCount = readExpression(first);

printf("\nSecond Expression:\n");

secondCount = readExpression(second);

printf("\nFirst Expression");

display(first, firstCount);

display(second, secondCount);

resultCount = addExpressions(firstCount, secondCount);

printf("\nResultant Expression:\n");

display(result, resultCount);

return 0;

}

The output of this code will also depend on the input that you give to the polynomials. For example, if you enter the following values:

First Expression: Number of terms: 3

Enter the coefficients and exponents in DESCENDING order

Coefficient: 5 Exponent: 3

Coefficient: 4 Exponent: 2

Coefficient: 2 Exponent: 0

Second Expression: Number of terms: 4

Enter the coefficients and exponents in DESCENDING order Coefficient: 3 Exponent: 4 Coefficient: 2

Exponent: 3 Coefficient: -4

Exponent: 2 Coefficient: 1

Exponent: 1

The output will be:

First Expression 5X^3+ 4X^2+ 2X^0+ Second Expression 3X^4+ 2X^3+ -4X^2+ 1X^1+ Resultant Expression: 3X^4+ 7X^3+ 0X^2+ 1X^1+ 2X^0+

You can also use an online compiler to run the code and see the output yourself.