LAB TASKS

**TASK 01:**

**Write a program to swap two rows of a 2D array using pointers. Declare a 3x3 integer array and initialize it with values of your choice. Create a function that takes the base address of the array, the indices of the rows to be swapped, and the number of columns as parameters. Use pointer arithmetic inside the function to swap the rows. After swapping, display the updated array.**

**Source code:**

#include<stdio.h>

swaping(int(\*arr)[3],int r1,int r2)

{

for(int c = 0; c < 3;c++)

{

int temp = \*(\*(arr+r1)+c);

\*(\*(arr+r1)+c)=\*(\*(arr+r2)+c);

\*(\*(arr+r2)+c) = temp;

}

}

int main()

{

int a[3][3] = {1,2,3,4,5,6,7,8,9};

printf("Array before swapping\n");

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

{

for(int j=0;j<3;j++)

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

printf("\n");

}

swaping(a,0,1);

printf("Array after swapping\n");

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

{

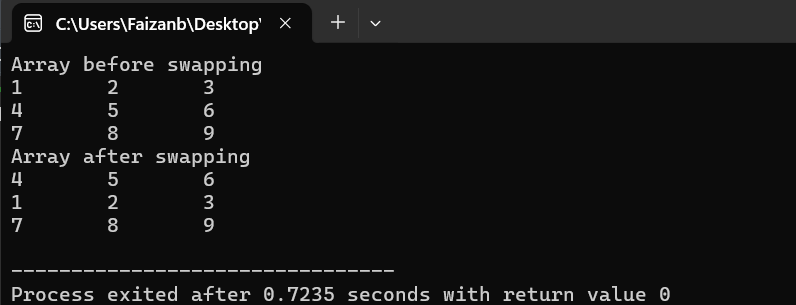
for(int j=0;j<3;j++)

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

printf("\n");

}

}



**TASK 02:**

**Write a program to dynamically allocate memory for two matrices, A and B, of sizes m x n and n x p, respectively. Populate the matrices with user-provided values, calculate their product matrix C, and print the result. Free all allocated memory afterward.**

**Source code:**

#include<stdio.h>

#include<stdlib.h>

int main()

{

int m,n,p;

printf("Enter m:");

scanf("%d",&m);

printf("Enter n:");

scanf("%d",&n);

printf("Enter p:");

scanf("%d",&p);

int \*arr = (int \*)malloc(m\*n\*sizeof(int));

int \*arr1 = (int \*)malloc(n\*p\*sizeof(int));

int \*arr2 = (int \*)malloc(m\*p\*sizeof(int));

printf("Enter %d elements of array 1\n",m\*n);

for(int i = 0;i<m;i++)

{

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

{

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

}

}

printf("Enter %d elements of array 2\n",n\*p);

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

{

for(int j =0;j<p;j++)

{

scanf("%d",arr1+i\*p+j);

}

}

for(int i = 0;i<m;i++)

{

for(int j =0;j<p;j++)

{

\*(arr2+i\*p+j) = 0;

}

}

for(int i = 0;i<m;i++)

{

for(int j = 0;j<p;j++)

{

for(int k = 0;k<n;k++)

{

\*(arr2+i\*p+j)+=\*(arr+i\*n+k) \* \*(arr1+k\*p+j);

}

}

}

printf("Product matrix\n");

for(int i = 0;i<m;i++)

{

for(int j = 0;j<p;j++)

printf("%d\t",\*(arr2+i\*p+j));

printf("\n");

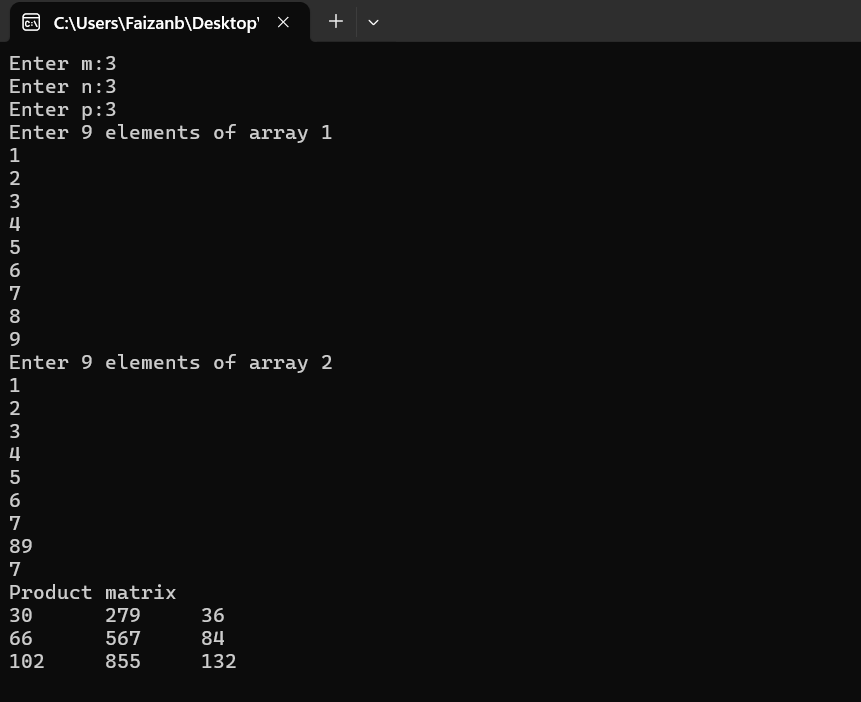
}

free(arr);

free(arr1);

free(arr2);

}



**TASK 03:**

**Write a program to search for a specific element in a 2D array using pointers. If the element is found, print its row and column indices.**

**Source code:**

#include<stdio.h>

#include<stdlib.h>

int main()

{

int r,c;

printf("Enter rows:");

scanf("%d",&r);

printf("Enter columns:");

scanf("%d",&c);

int \*arr = (int\*)malloc(r\*c\*sizeof(int));

int i,j;

printf("Enter %d elements of array\n",r\*c);

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

{

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

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

}

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

{

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

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

printf("\n");

}

int search;

int f = 0;

printf("Enter element to search:");

scanf("%d",&search);

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

{

for(j=0;j<3;j++)

{

if(\*(arr+i\*c+j)==search)

{

printf("Element is found at row %d and column %d",i,j);

f=1;

}

}

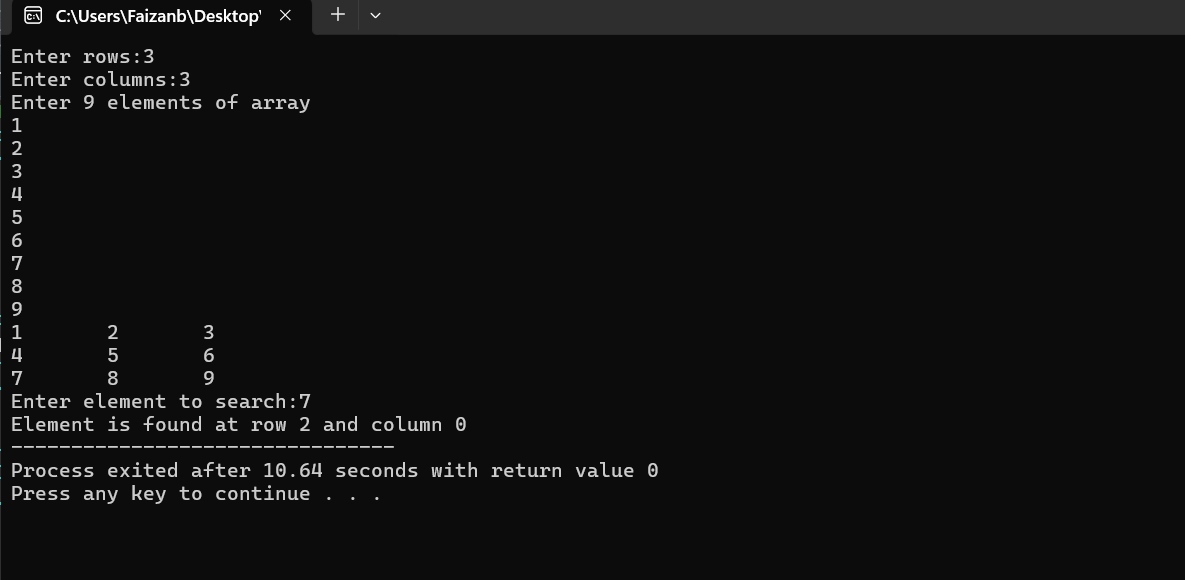
}

if(!f)

printf("Element not found");

free(arr);

}



**TASK 04:**

**Write a program to calculate the sum of all boundary elements of a 3x3 2D array using pointers. Ensure that only the first and last rows and columns are included in the sum.**

**Source code:**

#include<stdio.h>

int main()

{

int arr[3][3];

int i,j;

printf("Enter 9 elements of array\n");

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

{

for(j = 0;j<3;j++)

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

}

printf("Array\n");

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

{

for(j = 0;j<3;j++)

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

printf("\n");

}

int sum = 0;

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

{

for(j = 0;j<3;j++)

{

if(i == 0 || j==0 || i==2 || j==2)

{

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

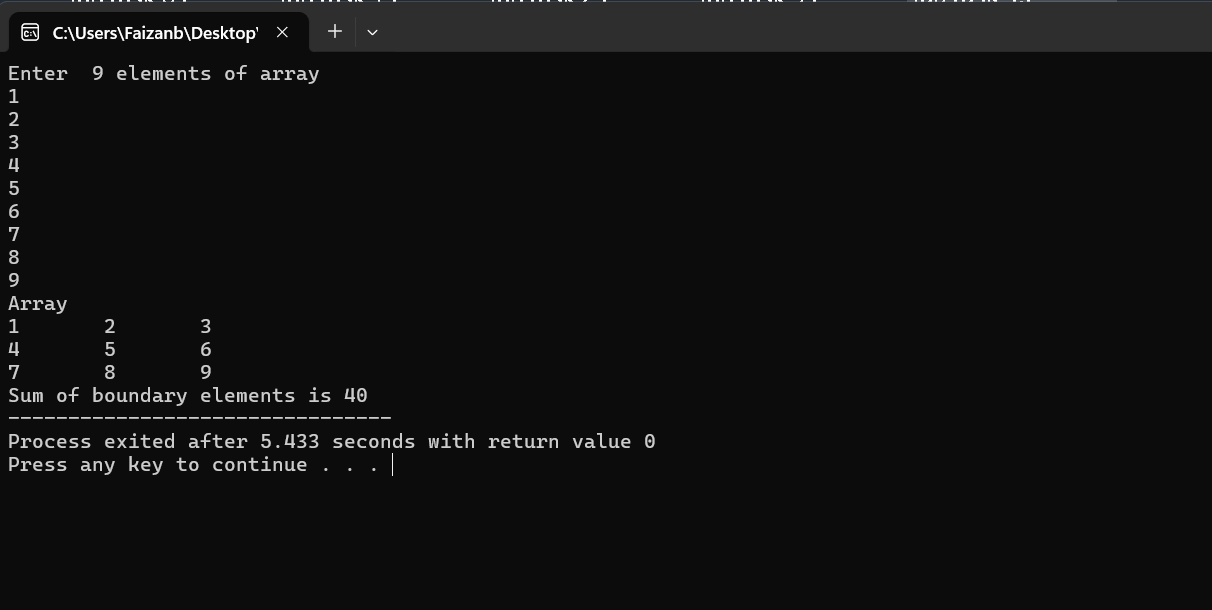
}

}

}

printf("Sum of boundary elements is %d",sum);

}



**TASK 05:**

**Write a program to dynamically allocate memory for an array of integers of size n entered by the user. Populate the array, double its size using realloc, add new elements, and display the updated array.**

**Source code:**

#include<stdio.h>

#include<stdlib.h>

int main()

{

int n ;

printf("Enter no of elements:");

scanf("%d",&n);

int \*arr = (int \*)malloc(n\*sizeof(int));

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

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

{

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

}

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

{

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

}

int n1 = n\*2;

arr=(int\*)realloc(arr,n1\*sizeof(int));

printf("\nEnter %d new elements\n",n1);

for(int i=0;i<n1;i++)

{

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

}

printf("Updated array:");

for(int i=0;i<n1;i++)

{

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

}

free(arr);

}

A screenshot of a computer

Description automatically generated

**TASK 06:**

**Write a program to dynamically allocate memory for an array of n strings, where each string can have a different length. Populate the strings with user input, sort them alphabetically, and display the sorted strings**

**Source code:**

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

void sort\_string(char \*\*str,int n)

{

char \*temp;

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

{

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

{

if(strcmp(str[i],str[j])>0)

{

temp = str[i];

str[i] = str[j];

str[j] = temp;

}

}

}

}

int main()

{

int n,len;

printf("Enter no of strings:");

scanf("%d",&n);

char \*\*str = (char\*)malloc(n\*sizeof(char\*));

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

{

char buffer[100];

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

scanf(" %[^\n]",buffer);

buffer[strcspn(buffer, "\n")] = 0;

str[i] = (char \*)malloc(strlen(buffer)+1\*sizeof(char));

strcpy(str[i],buffer);

}

sort\_string(str,n);

printf("Alphabetically sorted strings\n");

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

{

printf("%s\n",str[i]);

free(str[i]);

}

free(str);

}

