**Lab 12**

* **Question 1:**
* Source code:

#include <stdio.h>

#include <stdlib.h>

int main()

{

int n, i, sum=0;

printf("Enter the number of elements: ");

scanf("%d",&n);

int \*arr=(int \*)calloc(n , sizeof(int));

if(arr==NULL)

{

printf("Memory allocation failed!\n");

return 1;

}

printf("Input array elements:\n ");

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

{

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

sum+=arr[i];

}

printf("Sum is: %d",sum);

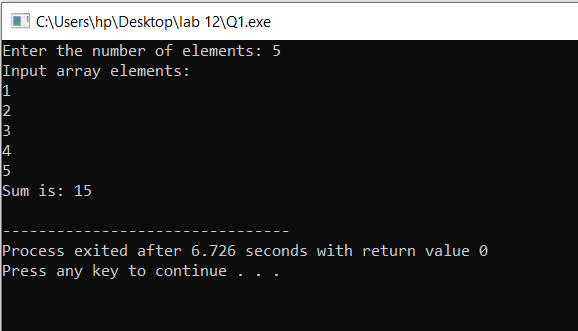
printf("\n");

free(arr);

return 0;

}

* Output:



* **Question 2:**
* Source code:

#include <stdio.h>

#include <stdlib.h>

int main()

{

int n, i, x=0, y=1, sum;

printf("Enter the number of elements: ");

scanf("%d",&n);

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

if(arr==NULL)

{

printf("Memory allocation failed!\n");

return 1;

}

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

{

printf("%d ", x);

arr[i]=x;

sum=x+y;

x=y;

y=sum;

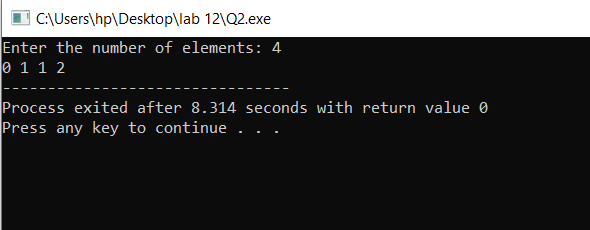
}

free(arr);

return 0;

}

* Output:



* **Question 3:**
* Source code:

#include <stdio.h>

#include <stdlib.h>

int main()

{

int rows, columns, i, j;

printf("Enter the number of rows: ");

scanf("%d",&rows);

printf("Enter the number of columns: ");

scanf("%d",&columns);

int \*arr=(int \*)calloc(rows\*columns , sizeof(int));

if(arr==NULL)

{

printf("Memory allocation failed!\n");

return 1;

}

printf("Input array elements:\n ");

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

{

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

{

scanf("%d",&arr[i \* columns + j]);

}

}

printf("Orignal matrix:\n");

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

{

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

{

printf("%d ",arr[i \* columns + j]);

}

printf("\n");

}

printf("\nTranspose:\n");

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

{

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

{

printf("%d ",arr[i\* columns + j]);

}

printf("\n");

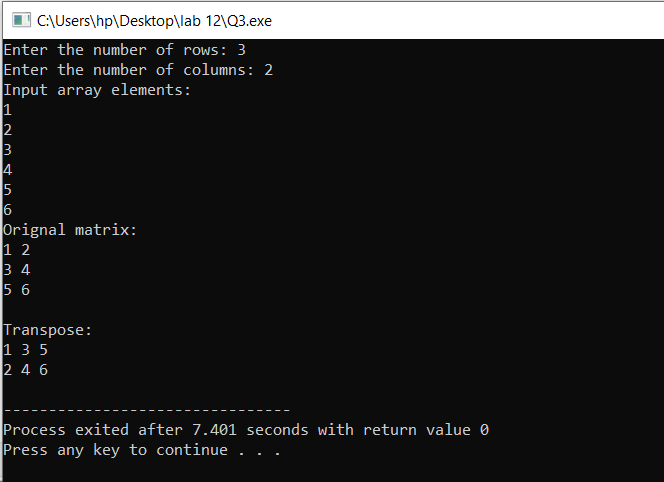
}

free(arr);

return 0;

}

* Output:



* **Question 4:**
* Source code:

#include <stdio.h>

#include <stdlib.h>

int main()

{

int n, i, found=0;

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

scanf("%d",&n);

char \*arr=(char \*)malloc((n+1)\* sizeof(char));

if(arr==NULL)

{

printf("Memory allocation failed!\n");

return 1;

}

printf("Enter a string: ");

scanf("%s",arr);

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

{

if(arr[i]!=arr[n-1-i])

{

found=0;

}

else

{

found=1;

}

}

if(found)

printf("Palindrome string");

else

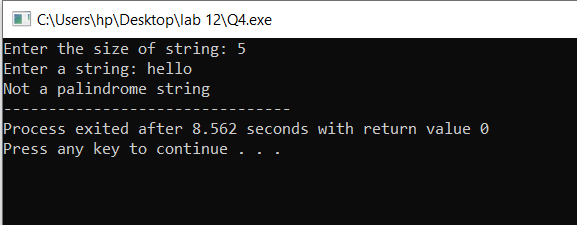
printf("Not a palindrome string");

free(arr);

return 0;

}

* Output:



* **Question 5:**
* Source code:

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int main()

{

int n, count=0, i;

printf("Enter the number of strings you want to enter: ");

scanf("%d", &n);

getchar();

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

if(str == NULL)

{

printf("Memory allocation failed");

return 1;

}

printf("Enter strings(\"END\" to stop):\n");

char arr[100];

while(count < n)

{

if(!fgets(arr, sizeof(arr),stdin))

{

printf("Error reading input");

break;

}

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

if(strcmp(arr,"END")==0)

{

break;

}

if(count == n)

{

n \*= 2;

char \*\*temp = realloc(str, n \* sizeof(char \*));

{

printf("Memory reallocation failed\n");

break;

}

str = temp;

}

str[count]= malloc(strlen(arr)+1);

if(str[count] == NULL)

{

printf("Memory reallocation failed");

return 1;

}

strcpy(str[count],arr);

count++;

}

printf("\nStrings:\n");

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

{

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

free(str[i]);

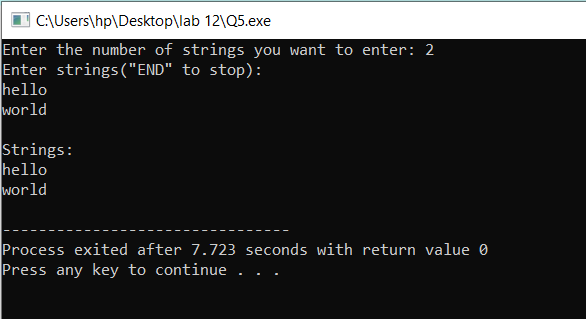
}

free(str);

return 0;

}

* Output:



* **Question 6:**
* Source code:

#include <stdio.h>

#include <stdlib.h>

int main()

{

int n, start, i, base=0;

printf("Enter starting number: ");

scanf("%d", &start);

printf("Enter number of elements: ");

scanf("%d",&n);

int \*arr= (int \*)calloc(n,sizeof(int));

printf("Range:\n");

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

{

arr[i]=start+base;

base++;

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

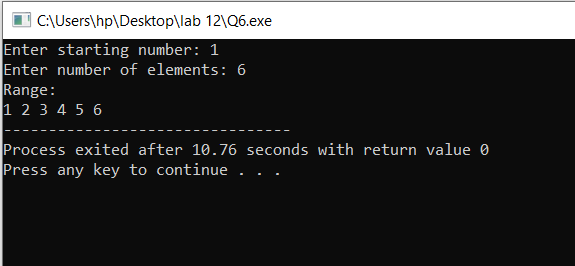
}

free(arr);

return 0;

}

* Output:



* **Question 7:**
* Source code:

#include <stdio.h>

#include <stdlib.h>

int main() {

int row1, row2, column1, column2, i, j, k, sum;

printf("Enter number of rows for matrix 1: ");

scanf("%d", &row1);

printf("Enter number of columns for matrix 1: ");

scanf("%d", &column1);

printf("Enter number of rows for matrix 2: ");

scanf("%d", &row2);

printf("Enter number of columns for matrix 2: ");

scanf("%d", &column2);

if (column1 != row2) {

printf("Matrix multiplication is not possible.\n");

return -1;

}

int \*arr1 = (int \*)malloc(row1 \* column1 \* sizeof(int));

int \*arr2 = (int \*)malloc(row2 \* column2 \* sizeof(int));

printf("Enter elements of matrix1:\n");

for (i = 0; i < row1; i++) {

for (j = 0; j < column1; j++) {

printf("value at [%d][%d]: ", i, j);

scanf("%d", &arr1[i \* column1 + j]);

}

}

printf("Enter elements of matrix2:\n");

for (i = 0; i < row2; i++) {

for (j = 0; j < column2; j++) {

printf("value at [%d][%d]: ", i, j);

scanf("%d", &arr2[i \* column2 + j]);

}

}

printf("Resultant matrix:\n");

for (i = 0; i < row1; i++) {

for (j = 0; j < column2; j++) {

sum = 0;

for (k = 0; k < column1; k++) {

sum += arr1[i \* column1 + k] \* arr2[k \* column2 + j];

}

printf("%d ", sum);

}

printf("\n");

}

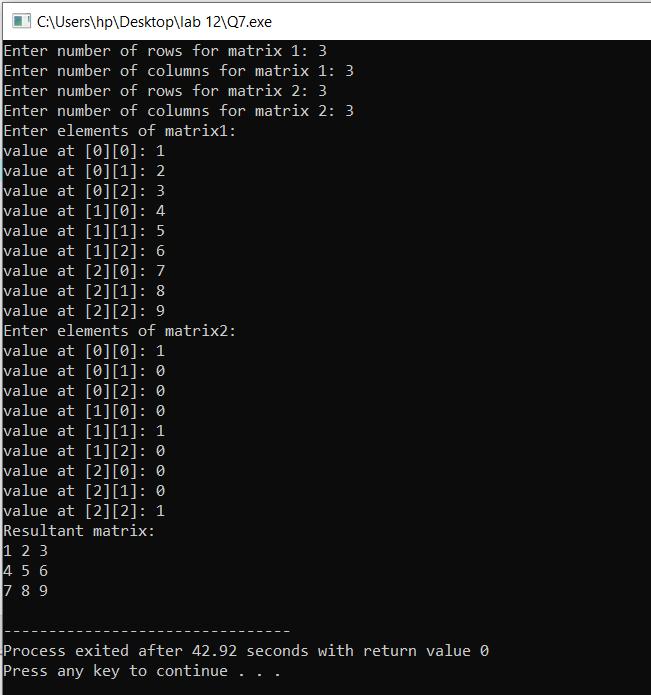
free(arr1);

free(arr2);

return 0;

}

* Output:



* **Question 8:**
* Source code:

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int main()

{

char \*ptr = malloc(1);

if (ptr == NULL)

{

printf("Initial memory allocation failed!\n");

return 1;

}

ptr[0] = '\0';

char arr[100];

int len1 = 0, i;

printf("Enter strings to concatenate (type 'STOP' to finish):\n");

while (1)

{

printf("Enter a string: ");

scanf("%s", arr);

if (strcmp(arr, "STOP") == 0)

{

break;

}

int len2 = strlen(arr);

char \*result = realloc(ptr, len1 + len2 + 1);

if (result == NULL)

{

printf("Memory allocation failed!\n");

free(ptr);

return 1;

}

ptr = result;

for (i = 0; i < len2; i++)

{

ptr[len1 + i] = arr[i];

}

len1 += len2;

ptr[len1] = '\0';

}

printf("\nConcatenated String: %s\n", ptr);

free(ptr);

return 0;

}

* Output:

