**Arrays and Pointers**

**Swaminathan navinashok**

**2019115126**

Pgm 7.1.1

C program to implement the following string manipulation functions using char pointers.

1. mystrlen()

int mystrlen(char \*s)

1. mystrcopy()

char\* mystrcopy(char \*dest, char \*source)

1. mystrcmp()

int mystrcmp(char \*s, char \*t)

1. mystrcat()

char\* mystrcat(char \*s1, char\*s2)

Pgm 7.1.2

### C program to implement matrix addition and matrix multiplication using two dimensional arrays.

### void read(int a[][], int nrows, int ncols);

### void sum(int a[][], int b[][], int c[][], int nrows, int ncols);

### void print( int c[][], int nrows, int ncols);

void main()

{

int a[10][10], b[10][10], c[10][10];

..

}

Pgm 7.1.3

### C program to implement matrix addition and matrix multiplication by declaring and passing two dimensional arrays as array of pointers (each pointer pointing to an array – dynamically allocate memory for each array).

### Use malloc() function to dynamically allocate memory.

### void read(int \*a[], int nrows, int ncols);

### void sum(int \*a[], int \*b[], int \*c[], int nrows, int ncols);

### void print( int \*c[], int nrows, int ncols);

void main()

{

int \*a[10], \*b[10], \*c[10];

..

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

{

a[i] = (int\*)malloc(ncols\*sizeof(int));

b[i] =

c[i] =

}

....

}

#include<stdio.h>

char \* mystrcat(char \*p1,char\* p2)

{ while(\*p1)

    {

        p1++;

    }

    while(\*p2)

    {

        \*p1 = \*p2;

        p2++;

        p1++;

    }

    \*p1 = '\0';

    return p1;

}

int main()

{

    char s1[100], s2[100];char \*p1;

    printf("\nstring1 \n: ");

   fgets(s1,100,stdin);

    printf("\n string2 \n: ");

 fgets(s2,100,stdin);

    p1=mystrcat(s1,s2);

    printf("\n %s ", s1);

    return 0;

}

#include<stdio.h>

int slen(char\*s)

{

   int i = 0;

   while (\*s!='\0')

    { s++;i++; }

   return i;

}

void main() {

   char str[20];

   int length;

 printf("string : ");

fgets(str,20,stdin);

   length = slen(str);

   printf("\n lengt of %s: %d \n", str, length);

}

#include<stdio.h>

int mystrcmp(char \*first, char \*second)

{

   while(\*first==\*second)

   {

      if ( \*first == '\0' || \*second == '\0' )

         break;

      first++;

      second++;

   }

   if( \*first == '\0' && \*second == '\0' )

      return 0;

   else

      return -1;

}

int main()

{

    char s1[100], s2[100], r;

    printf("string\n");

 fgets(s1,100,stdin);

    printf("string\n");

    fgets(s2,100,stdin);

    r= mystrcmp(s1, s2);

    if ( r == 0 )

       printf("strings are same.\n");

    else

       printf("strings are not equal.\n");

    return 0;

}

#include<stdio.h>

char\* stcpy(char \*s1, char \*s2)

{

    while(\*s2!='\0')

   { \*s1 = \*s2;s1++; s2++;}

      \*s1 = '\0';

      return s1;

}

int main()

{

    char s1[100], s2[100]; char\* p;

    printf(" string\n");

    fgets(s2,100,stdin);

    p=stcpy(s1, s2);

    printf("%s \n", s1);

    return 0;

}

#include <stdio.h>

// function to get matrix elements entered by the user

void read(int a[][10], int nrow, int ncol) {

   printf("\nEnter elements: \n");

   for (int i = 0; i < nrow; ++i) {

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

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

         scanf("%d", &a[i][j]);

      }

   }

}

void sum(int a[][10],

                      int b[][10],

                      int c[][10],

                      int nrow, int ncol) {

   for (int i = 0; i < nrow; ++i)

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

            c[i][j]=a[i][j]+b[i][j];

            if (j == ncol - 1) {

                printf("\n\n");

            }

        }

}

// function to display the matrix

void print(int c[][10], int nrow, int ncol) {

   printf("\nOutput Matrix:\n");

   for (int i = 0; i < nrow; ++i) {

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

         printf("%d  ",c[i][j]);

         if (j == ncol - 1)

            printf("\n");

      }

   }

}

int main() {

   int a[10][10], b[10][10],c[10][10], nrow, ncol;

   printf("Enter rows and column for the first matrixes: ");

   scanf("%d %d", &nrow, &ncol);

   // get elements of the first matrix

   read(a, nrow, ncol);

   // get elements of the second matrix

   read(b, nrow, ncol);

   // multiply two matrices.

  sum(a,b,c, nrow, ncol);

   // display the result

   print(c, nrow, ncol);

   return 0;

}

#include<stdio.h>

#include<stdlib.h>

#define maxc 50

void read(int \*a[maxc], int r, int c)

{

    for(int i = 0; i < r; ++i)

    {

        a[i] = (int \*)malloc(sizeof(int)\*c);

        for(int j = 0; j < c; ++j)

            scanf("%d", \*(a + i)+j);

    }

}

void printarr(int \*a[maxc], int r, int c)

{

    for(int i = 0; i<r; i++)

    {

        for(int j = 0; j < c; j++)

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

        printf("\n");

    }

}

void m\_add(int \*a[maxc], int \*b[maxc], int r, int c)

{

    int \*C[maxc];

    for(int i = 0; i < r; ++i)

    {

        C[i] = (int \*) malloc(sizeof(int)\*c);

        for(int j = 0; j < c; ++j)

            \*(\*(C+i)+j) = \*(\*(a+i)+j) + \*(\*(b+i)+j);

    }

    printf("A + B :\n");

    printarr(C, r, c);

}

int main()

{

    int \*a[maxc];int \*b[maxc];

    int r1, c1, r2, c2;

    printf("rows in A: "); scanf("%d", &r1);

    printf(" coulmns in A: "); scanf("%d", &c1);

    printf("elemnts of A:\n");

    read(a, r1, c1);

    printf("rows in B: "); scanf("%d", &r2);

    printf("coulmns in B: "); scanf("%d", &c2);

    printf("elemnts of B:\n");

    read(b, r2, c2);

        m\_add(a, b, r1, c1);

   return 0;

}

Outputs:



