Command Line Arguments and Arrays

Course: Introduction to Programming and Data Structure

Laltu Sardar

Institute for Advancing Intelligence (IAI),
TCG Centres for Research and Education in Science and Technology (TCG Crest)

tcg crest
Inventing Harmonious Future

Sep 05, 2022

Command Line Arguments

Why inputs from command line

- Another form of input
- Useful when you want to control your program from outside.
- To override defaults and have more direct control over the application

Example:

```
1  int main(int argc, char *argv[]) {
2    /* ... */
3  }
```

or

```
1  int main(int argc, char **argv) {
2    /* ... */
3  }
```

tcg crest
Inventing Harmonious Future

```c
// Program to compute average of two float variables
#include<stdio.h>
#include<stdlib.h> //that contains atof

float average(float a, float b){
    return ((a+b)/2.0);
}
int main(int argc, char *argv[]){
    float a, b, avg;
    if (argc==3){
        a = atof(argv[1]); //converting string to float
        b = atof(argv[2]);
    }else{
        scanf("%f %f", &a, &b);  // taking input from terminal
    }
    avg = average(a, b);     //Compauting avarage
    printf("%.2f",avg); //writing on terminal
    return 0;
}
```

tcg crest
Inventing Harmonious Future

```c
// Program to compute average of two float variables
#include<stdio.h>
#include<stdlib.h> //that contains atof

float average(float a, float b){
    return ((a+b)/2.0);
}
int main(int argc, char *argv[]){
    float a, b, avg;
    if (argc==3){
        a = atof(argv[1]); //converting string to float
        b = atof(argv[2]);
    }else{
        scanf("%f %f", &a, &b);  // taking input from terminal
    }
    avg = average(a, b);     //Compauting avarage
    printf("%.2f",avg); //writing on terminal
    return 0;
}
```

- argc (ARGument Counter): is The number of command-line arguments passed. It includes the name of the program
- argv (ARGument Vector): An array of strings pointers listing all the arguments.
- argv[0] is the name of the program , After that till argv[argc-1] every element is command-line arguments.
- Only strings can be taken from command line.

tcg crest
Inventing Harmonious Future

1-Dimensional Array

String: Array of Characters

- `char name[] = "crest" ;`
- `char *name = "crest";`

Integer array

- `int val[] = { 1, 2, 3,4 } ;` // declaration and initialization
- `int *val = { 1, 2, 3, 4};`

Problems

1. Find concatenation of two strings
2. find the number of appearances of a sub-string in a string
3. Replace a specific sub-string of a string with another sub-string

Multi-Dimensional Array

- `char *names[] = {"Soumya", "Prabal", "Rajani"};`
- How the memories are allocated for above strings?
- How argc, argvs are allocated. (main(int argc, char *argv[]))

Matrix

- `int A[n][m];`
- `int *A = (int *)malloc(n*sizeof(int));` instead A[n]
- `int **A = (int **)malloc(n*sizeof(int *));`

```
for (i =0 ; i< n ; i++)
{ int *A = (int *)malloc(m*sizeof(int)); }
```

- Accessing element: `A[i][j];`

tcg crest
Inventing Harmonious Future

Multi-Dimensional Array

- `char *names[] = {"Soumya", "Prabal", "Rajani"};`
- How the memories are allocated for above strings?
- How argc, argvs are allocated. (main(int argc, char *argv[]))

Matrix

- `int A[n][m];`
- `int *A = (int *)malloc(n*sizeof(int));` instead A[n]
- `int **A = (int **)malloc(n*sizeof(int *));`

```
for (i =0 ; i< n ; i++)
{ int *A = (int *)malloc(m*sizeof(int)); }
```

- Accessing element: `A[i][j];`

Benefits of malloc over A[n][m]?

- Low rate of failure for large dimension.

tcg crest
Inventing Harmonious Future

Multi-Dimensional Array

Class works

1. Define your own `my_Malloc_int(n, m)`, with error message, that returns integer matrix of size $n \times m$
2. Declare two matrices A and B of size $n \times m$. Allocate integer memory for them. take input from a input file.
3. define `add_matrix(A,B,n,m)` that adds two matrices.
4. define `add_matrix(A,B,n,m)` that multiply two matrices.

tcg crest
Inventing Harmonious Future

Play with Matrices

Write a program that add two matrices

1. Step 1: Write function "matrix_add(A, B, m, n)" that takes input pointers to the two matrices A, B and outputs another matrix C.
2. Step2: write a function that prints elements of a matrix. Then print the result matrix

tcg crest
Inventing Harmonious Future

Play with Matrices

Write a program that add two matrices

1. Step 1: Write function "matrix_add(A, B, m, n)" that takes input pointers to the two matrices A, B and outputs another matrix C.
2. Step2: write a function that prints elements of a matrix. Then print the result matrix

Write a program that subtract one matrix from ther

1. use the same program.
2. Homework

tcg crest
Inventing Harmonious Future

Play with Matrices

Write a program that add two matrices

1. Step 1: Write function "matrix_add(A, B, m, n)" that takes input pointers to the two matrices A, B and outputs another matrix C.
2. Step2: write a function that prints elements of a matrix. Then print the result matrix

Write a program that subtract one matrix from ther

1. use the same program.
2. Homework

Write a program that multiply two matrices

1. use the same program.