محاضرات العملي لمقرر

مقدمة في البرمجة

ITGS122

الدرس العملي رقم (4)

اعداد الأستاذة : هبة الهويجي

- **المصفوفات Arrays في لغة البرمجة C:**

هي مجموعة محددة من البيانات في غالب الأمر تكون بيانات ذات صلة ويجب أن تنتمي هذه البيانات الى نفس النوعData-Type ، تسمى هذه البيانات بـ عناصر المصفوفة أو Array Elements، ويتم تخزين هذه البيانات أو العناصر في الذاكرة بشكل متجاور أو متلاصق.

قد تحتوي المصفوفة على مصفوفات أخرى من البيانات وفي هذه الحالة تعتبر مصفوفة متعددة الأبعاد أو Multi-Dimensional Array، وتعد المصفوفة ثنائية الأبعاد Two-Dimensional Arrayهي الأكثر إستخداما لحل مسائل المصفوفات الرياضية والمعروفة بـ.Matrix .

تسمى المصفوفة بأحادية البعد أو One-Dimensional Arrayإذا ما كانت عناصر هذه المصفوفة مرتبة بشكل قائمة أو بمعنى أدق أن عناصر هذه المصفوفة ليست في حد ذاتها مصفوفات أخرى.

الشكل العام أو القاعدة العامة لإعلان المصفوفة في لغة البرمجة سي تكون كالتالي

```
Data-Type ArrayName[PositiveIntegerNumber];
```

حيث يتم أولا كتابة نوع البيانات الممثل لعناصر المصفوفة، ثم يتم إختيار إسم متغير للمصفوفة بحيث يتم إتباع القواعد الخاصة لإختيار إسم المتغير في لغة البرمجة سي كما تعرفنا عليها سابقا، ثم يتم إستخدام الأقواس المربعة أو Square Brackets وبداخلهم يتم كتابة رقم صحيح موجب أكبر من الصفر لكي يحدد عدد عناصر المصفوفة ويتم إنهاء الجملة بعلامة الفاصلة المنقوطة.Semicolon

Program to find sum of array elements using loops

```c
#include <stdio.h>
int main()
{
  int arr[100],size,sum=0;

  printf("Enter size of the array: ");
  scanf("%d",&size);

  printf("Enter the elements of the array: ");
  for(int i=0; i<size; i++)
  {
    scanf("%d",&arr[i]);
  }
  //calculating sum of entered array elements
  for(int i=0; i<size; i++)
  {
    sum+=arr[i];
  }
  printf("Sum of array elements is: %d",sum);

  return 0;
}
```

Output

```
Enter size of the array: 5
Enter the elements of the array: 10 20 30 40 50
Sum of array elements is: 150
```

C Program to print largest element of an array

```c
#include <stdio.h>

int main() {
    int n, i;

    // Prompt user to enter the number of elements in the array
    printf("Enter the number of elements in the array: ");
    scanf("%d", &n); //store the entered value in variable n

    // Declare an array of size n to store array elements
    int array[n];

    // Prompt user to enter elements of the array
    printf("Enter the elements of the array:\n");
    for(i = 0; i < n; i++) {
        printf("Element %d: ", i + 1);
        scanf("%d", &array[i]);
    }

    // Initialize the largest variable with first element
    int largest = array[0];

    /* Here we are comparing largest with
     * all other elements of array. If any element is found
     * greater than largest, then value of largest is updated to
     * to contain this element. This way whole array is traversed
     * and the largest variable contains largest element of array
     */
    for(i = 1; i < n; i++) {
        if(array[i] > largest) {
            largest = array[i];
        }
    }

    // Print the largest element
    printf("The largest element in the array is: %d\n", largest);

    return 0;
}
```

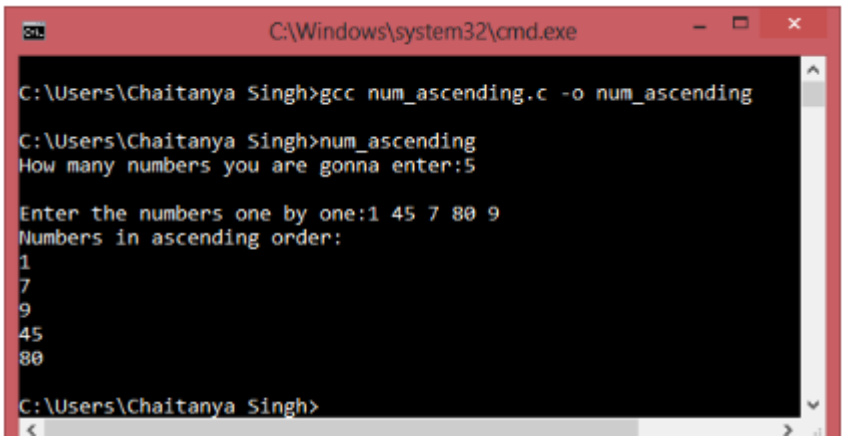
Output

```
Enter the number of elements in the array: 5
Enter the elements of the array:
Element 1: 3
Element 2: 7
Element 3: 2
Element 4: 9
Element 5: 5
The largest element in the array is: 9
```

C Program to arrange numbers in ascending order

```c
/*
 * C program to accept numbers as an input from user
 * and to sort them in ascending order.
 */
#include <stdio.h>

void sort_numbers_ascending(int number[], int count)
{
   int temp, i, j, k;
   for (j = 0; j < count; ++j)
   {
      for (k = j + 1; k < count; ++k)
      {
         if (number[j] > number[k])
         {
            temp = number[j];
            number[j] = number[k];
            number[k] = temp;
         }
      }
   }
   printf("Numbers in ascending order:\n");
   for (i = 0; i < count; ++i)
      printf("%d\n", number[i]);
}
void main()
{
   int i, count, number[20];

   printf("How many numbers you are gonna enter:");
   scanf("%d", &count);
   printf("\nEnter the numbers one by one:");

   for (i = 0; i < count; ++i)
      scanf("%d", &number[i]);

   sort_numbers_ascending(number, count);
}
```

Output



Program to find the size of an array

```c
#include <stdio.h>
int main()
{
    double arr[] = {11, 22, 33, 44, 55, 66};
    int n;

    /* Calculating the size of the array with this formula.
     * n = sizeof(array_name) / sizeof(array_name[0])
     * This is a universal formula to find number of elements in
     * an array, which means it will work for arrays of all data
     * types such as int, char, float etc.
     */
    n = sizeof(arr) / sizeof(arr[0]);
    printf("Size of the array is: %d\n", n);
    return 0;
}
```

Output

```
Size of the array is: 6
```

Printing Two-Dimensional Array

```c
#include <stdio.h>

int main(){

   int i, j;
   int x[4][4] = {
      {1, 2, 3, 4},
      {11, 22, 33, 44},
      {9, 99, 999, 9999},
      {10, 20, 30, 40}
   };

   // outer loop
   for (i=0; i<=3; i++){

      // inner loop
      for(j=0; j <= 3; j++){
         printf("%5d", x[i][j]);
      }
      printf("\n");
   }
   return 0;
}
```

Output

```
    1    2    3    4
   11   22   33   44
    9   99  999 9999
   10   20   30   40
```

In the following example, we are declaring, initializing, and printing an array of string

```c
#include <stdio.h>

int main (){

   char langs [10][15] = {
      "PYTHON", "JAVASCRIPT", "PHP",
      "NODE JS", "HTML", "KOTLIN", "C++",
      "REACT JS", "RUST", "VBSCRIPT"
   };

   for (int i = 0; i < 10; i++){
      printf("%s\n", langs[i]);
   }

   return 0;
}
```

Output

```
PYTHON
JAVASCRIPT
PHP
NODE JS
HTML
KOTLIN
C++
REACT JS
RUST
VBSCRIPT
```