TUTORIAL 5

Tutorial Task(s)

1. Explain the keywords and key terms: a) Array identifier b) Array Initialization c) Scalar variable d) Scanset e) Size of array f) string compare g) string copy h) string manipulation

a) Array identifier: A name assigned to an array.

b)Array initialization: The procedure of assigning numerical value or character to each element of an array.

c) A scalar variable is a single variable whose stored value is an atomic type

d) Scanset: It is a conversion specifier that allows the programmer to specify the set of characters that are (or are not) acceptable as part of the string.

e) Size of array: The number of elements in an array.

f) String compare: A kind of string manipulation where two strings are compared to primarily find out whether they are similar or not.

g)String copy: A kind of string manipulation where one string is copied into another.

h)String manipulation: Carrying out various operations like comparing, appending, copying, etc. among strings

2. Explain the concept of array?

   An array is a data structure, which can store a fixed-size collection of elements of the same data type.

3. Mention the reason" can array index can be negative?

Array index cannot be negative as it ranges from 0 to (size of array - 1) only. If it's negative then compiler will show error.

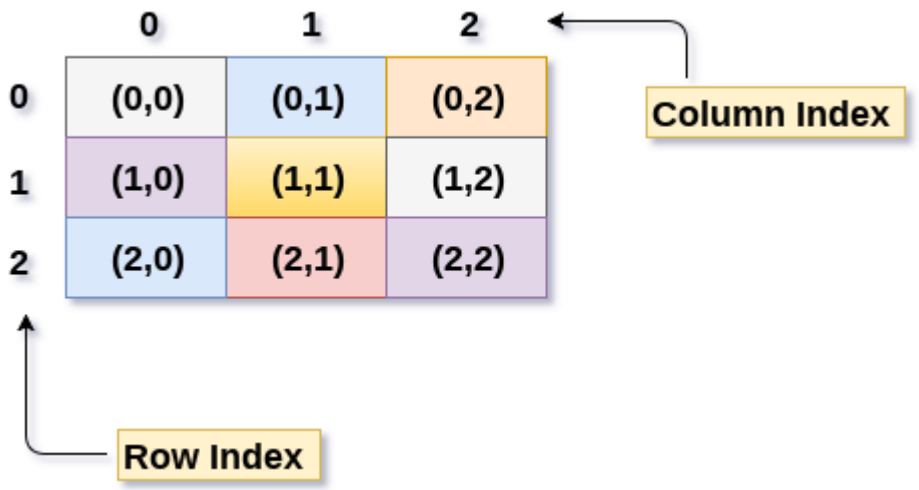
4. Why it is necessary to give the size of an array in array declaration?

We need to give the size of the array because the complier needs to allocate space in the memory which is not possible without knowing the size. Compiler determines the size required for an array with the help of the number of elements of an array and the size of the data type present in the array.

5. Explain one -dimensional , two dimensional and strings?

- One dimensional array- there will be a single subscript or index whose value refers to the individual array element which ranges from 0 to (n–1), where n is the total number of elements in the array.
- The elements of a 2 dimensional array are arranged in rows and columns, and has 2 square brackets denoting number of rows and columns respectively.
- String : One-dimensional array of characters that contain a NUL at the end

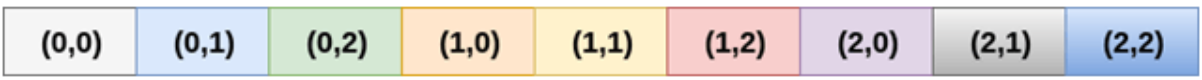
6. Demonstrate the storage of two-dimensional arrays in a memory with help of diagram?



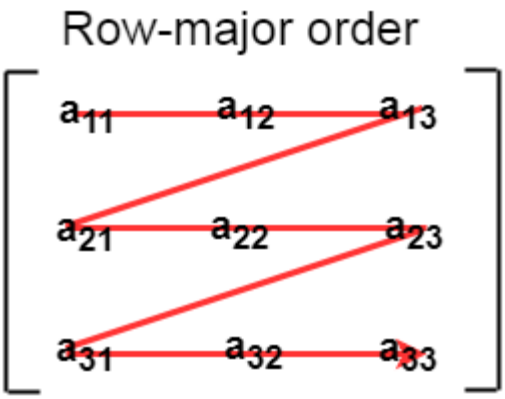
There are two main techniques of storing 2D array elements into memory

1. Row Major ordering

In row major ordering, all the rows of the 2D array are stored into the memory contiguously. Considering the array shown in the above image, its memory allocation according to row major order is shown as follows.

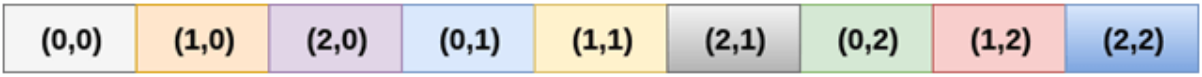


First, the 1st row of the array is stored into the memory completely, then the 2nd row of the array is stored into the memory completely and so on till the last row.

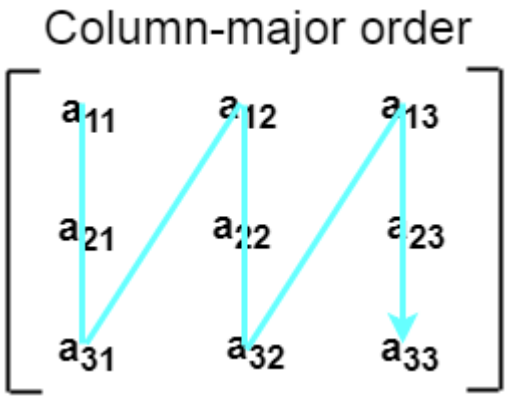


2. Column Major ordering

According to the column major ordering, all the columns of the 2D array are stored into the memory contiguously. The memory allocation of the array which is shown in in the above image is given as follows.



First, the 1st column of the array is stored into the memory completely, then the 2nd row of the array is stored into the memory completely and so on till the last column of the array.



7. Why does array subscripts start at 0 instead of 1?

The first element of the array is exactly contained in the memory location that array refers (0 elements away), so it should be denoted as array[0]

8. What are the characteristics of array?

1) An array is a derived data type, which is defined using basic data types like int, char, float.

2) Array elements are stored in contiguous memory blocks/subsequent memory blocks in primary memory.

3) Array name represents its base address. The base address is the address of the first element of the array.

4) Array's index starts with 0 and ends with N-1. Here, N stands for the number of elements.

5) Only constants and literal values (an integer value like 5, 10, 12,...) can be assigned the number of elements in an array.

9. Why do we have a null character ('\0 or NUL) at the end of string?

In C language, string is an array of char type. When followed by a number, the escape sequence represents the ascii character (in octal) associated with the number, thus \0 means the character with ascii value zero, also known as NUL.

10.If a string str contains string literal " Coimbatore institute of technology", then is it legal to print the string using statement printf(str);?

No as proper syntax should be used for printf. Proper syntax is printf("%s",str);

Practice!!!

1. Write a program that perform the following:

The user inputs a number and then enters a series of number from 1 to that number.Your program should determine which number (or numbers) is missing or duplicated in the series, if any, For example, if

the user entered 5 as the initial number and then entered the following sequences, the result should be as shown.

Input Sequence Output

---------- ----------

1 2 3 4 5 Nothing bad

However if 7 were the highest number, the user would see the results on the right for the following number entries:

Input Sequence Output

---------- ----------

1 2 3 4 5 Missing 6

Missing 7

And if 10 were the highest number and the user entered the numbers shown on the left, note the list of missing and duplicate numbers:

Input Sequence Output

---------- ----------

1 2 4 7 4 4 5 10 8 2 6 Duplicate 2 (2 times)

Missing 3

Duplicate 4 (3times)

Missing 9

The program should check the highest number that the user input to ensure that it does not exceed the size of any array you might be using for storage

```
#include <stdio.h>

int main()
{
int a[100];
int s,c;
int b=0;

printf("Enter last number of your series:\n");
scanf("%d",&c);

printf("Enter how many numbers you want to check:\n");
scanf("%d",&s);

printf("Type the numbers you want to check\n");
for (int i=0;i<s;i++)
scanf("%d",&a[i]);

for (int x=1;x<=c;x++)
{
for (int y=0;y<s;y++)
if (a[y]==x)
b++;
if (b>=2)
```

```
printf("Duplicate %d :(%d) time's\n",x,b);
if (b==0)
printf("%d is missing.\n",x);
b=0;
}

   return 0;
}
```

2. Write a program for following problems related to string:

a) Read a word and rewrite it in alphabetical order

```
#include<stdio.h>

 int main(){

      char str[100],temp;
       int i,j;

       printf("Enter the string :");
       scanf("%s",str);

       printf("%s in alphabetical order is: ",str);
       for(i=0;str[i];i++)
       {
```

```
            for(j=i+1;str[j];j++)
            {
                if(str[j]<str[i])
                {
                    temp=str[j];
                    str[j]=str[i];
                    str[i]=temp;
                }
            }
        }
        printf("%s\n",str);
        return 0;
    }
```

b) Take the name of person as input and print the first letter of the first name and middle name(if any) and the title as it is, e.g., Raj Kumar Santhoshi as R.K.Santhoshi

```
#include<stdio.h>
#include <string.h>
#include <ctype.h>

 int main(){

     char fir[50], mid[50], las[50];
```

```
int i, j;

printf("Enter first name: ");
scanf("%s",fir);

printf("Do you have middle name? Press: 0- yes/1- no:");
scanf("%d", &j);

if(j==0){
printf("Enter middle name: ");
scanf("%s",mid);
}

printf("Enter last name: ");
scanf("%s",las);

printf("The title is: ");

  printf("%c.",toupper(fir[0]));
  if(j==0){
  printf("%c.",toupper(mid[0]));
  }
  printf(" %s",las);
```

```
    return 0;
}
```

c) To find a string within a sentence and replace it with another string

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
void replaceWordInText(const char *text, const char *oldWord,
const char *newWord) {
  int i = 0, cnt = 0;
  int len1 = strlen(newWord);
  int len2 = strlen(oldWord);
  for (i = 0; text[i] != '\0'; i++) {
    if (strstr(&text[i], oldWord) == &text[i]) {
      cnt++;
      i += len2 - 1;
    }
  }
  char *newString = (char *)malloc(i + cnt * (len1 - len2) + 1);
  i = 0;
  while (*text) {
    if (strstr(text, oldWord) == text) {
```

```
        strcpy(&newString[i], newWord);
        i += len1;
        text += len2;
      }
      else
      newString[i++] = *text++;
   }
   printf("New String: %s\n", newString);
}
int main() {
   char str[] = "fox fox hi";
   char c[] = "fox";
   char d[] = "hi";
   char *result = NULL;

   printf("Original string, string to be removed, and string to be
replaced: %s\n%s\n%s\n", str,c,d);
   replaceWordInText(str, c, d);
   return 0;
}
```

d) Delete a word from a sentence. Note that the word may appear any number of times.

```
#include <stdio.h>
```

```
#include <string.h>
#include <stdlib.h>

int main()
{
    char str[64];
    char word[16];
    char words[6][16];

    int i = 0;
    int j = 0;
    int k = 0;
    int l1 = 0;
    int l2 = 0;

    printf("Enter string: ");
    scanf("%[^\n]s", str);

    printf("Enter word to be deleted: ");
    scanf("%s", word);

    while (str[i] != 0) {
        if (str[i] == ' ') {
            words[k][j] = '\0';
```

```
        k++;
        j = 0;
    }
    else {
        words[k][j] = str[i];
        j++;
    }
    i++;
}
words[k][j] = '\0';

j = 0;
for (i = 0; i < k + 1; i++) {
    if (strcmp(words[i], word) == 0)
        words[i][j] = 0;
}

j = 0;
printf("Result is:\n");
for (i = 0; i < k + 1; i++) {
    if (words[i][j] == 0)
        continue;
    else
        printf("%s ", words[i]);
```

```
    }
    printf("\n");

    return 0;
}
```