بِسْمِ ٱللَّهِ ٱلرَّحْمَٰنِ ٱلرَّحِيمِ

# C language

- Syntax

```
#include <stdio.h>
int main()
{
      printf("Hello, world!\n");
      return 0;
}
```

- Comments

  to coment only one line use //
  to comment multiple lines use /* pass text here */ (/* */)

- Variables
  For numbers like (-1, -2, 1, 2, 3, ..) use int
  For numbers like (-1.2, -2.54, 1.6, 2.12, 3.4545, ..) use float or double
  For charachters use char

- Data types
  - Basic format specifiers

| Format specifier | Data type |
|---|---|
| %d or %i | int |
| %f | float |
| %lf | double |
| %c | char |
| %s | used for strings |

- Constants
  - Const int myAge = 21; // myAge will always be 21
  - myAge = 10 // error : assignment of read only variables “myAge”

- Operators
  - Addition +
  - Substraction –
  - Multiplication *
  - Division /
  - Modulus %
  - Increment ++
  - Decrement –

- Output
  - Print a text

```c
#include <stdio.h>
int main()
{
	printf("Hello, world!\n");
	return 0;
}
```

  - Print array of integers

```c
#include <stdio.h>
int main()
{
	int array[] = {1, 2, 3, 4, 5};
	for (int i = 0; i < 5; i++)
		printf("%d, ", array[i]);
	return 0;
}
```

  - Print array of characters

```c
#include <stdio.h>
int main() {
	char array[][20] = {"Muhammed", "ElIdrissi", "Nada", "Jouidi"};
	printf("%s\n", array[0]);
	printf("%s\n", array[1]);
	printf("%s\n", array[2]);
	printf("%s\n\n", array[3]);
	for (int i = 0; i < 4; i++)
		printf("%s\n", array[i]);
	return 0;
}
```

  - New lines

- Input
  - Get an integers

```c
#include <stdio.h>
int main()
{
	int integer;
	scanf("%d", &integer);
	return 0;
}
```

  - Get array of numbers

```c
#include <stdio.h>
int main()
{
	int array[] = {1, 2, 3, 4, 5};
	for (int i = 0; i < 5; i++)
		scanf("%d, ", &array[i]);
	return 0;
}
```

  - Get a squence of characters

```c
#include <stdio.h>
int main() {
    char str[20];
    scanf("%[^\n]", &str);
    printf("%s", str);

    char text[20];
    gets(text);
    printf("%s", text);
    return 0;
}
```

  - Get an array of characters

```c
#include <stdio.h>
int main() {
    int size;
    printf("Enter size: ");
    scanf("%d", &size);
    char array[size][20];
    for (int i = 0; i < size; i++)
        scanf("%s\n", &array[i]);

    printf("\n-----------------\n");
    for (int i = 0; i < size; i++)
        printf("%s\n", array[i]);
    return 0;
}
```

- If .. else

```c
#include <stdio.h>
int main(){
    int a, b;
    a = 2;
    b = 1;
    if (a > b)
        printf("a is greater than b\n");
    else if (a < b)
        printf("b is greater than a\n");
    else
        printf("a is equal to b\n");
}
```

- Switch

```c
#include <stdio.h>
int main() {
     int choice;
     printf("Choice: ");  scanf("%d", &choice);
     switch (choice)
     {
     case 1:
          printf("This message displays number 1\n");
          break;
     case 2:
          printf("This message displays number 2\n");
          break;
     case 3:
          printf("This message displays number 3\n");
          break;
     default:
          printf("This message contains no number!\n");
          break;
     }
}
```

- While loop

```c
#include <stdio.h>
int main()
{
     int x = 7;
     while (x < 10)
     {
          printf("This is a while loop!\n");
          x++;
     }
}
```

- Do while loop

```c
#include <stdio.h>
int main()
{
     int x = 7;
     do
     {
          printf("This is a while loop!\n");
          x++;
     } while (x < 10);
}
```

- For loop

```c
#include <stdio.h>
int main()
{
     for (int i = 0; i < 5; i++)
          printf("message %d\n", i);
}
```

- **Break / continue**

break

```
#include <stdio.h>
int main()
{
    int i;
    for (i = 0; i < 5; i++)
        if (i == 4)
            break;
    printf("%d\n", i);
}
```

continue

```
#include <stdio.h>
int main()
{
    int i;
    for (i = 0; i < 5; i++)
        if (i == 4)
            continue;
    printf("%d\n", i);
}
```

- *Arrays*

  *Integers*

### Declaring and initializing array variables

```c
int array[] = {1, 2, 3, 4, 5};
```

### Input and output

```c
#include <stdio.h>
int main()
{
     int integer;
     scanf("%d", &integer);
     // input
     int array[] = {1, 2, 3, 4, 5};
     for (int i = 0; i < 5; i++)
          scanf("%d, ", &array[i]);

     // output
     int array[] = {1, 2, 3, 4, 5};
     for (int i = 0; i < 5; i++)
          printf("%d, ", array[i]);

     return 0;
}
```

### Size of the arrays using sizeof()

```c
int size = sizeof(array) / sizeof(array[0]);
```

### Sum and product of array numbers

```c
#include <stdio.h>
#include <stdlib.h>
int main()
{
     int sum = 0, product = 1;
     int array[] = {1, 2, 3, 4, 5};
     int size = sizeof(array) / sizeof(array[0]);
     for (int i = 0; i < size; i++)
     {
          sum += array[i];
          product *= array[i];
     }
     printf("Sum = %d\n", sum);
     printf("Product = %d\n", product);

     return 0;
}
```

Find minimum & maximum numbers in the array

```c
#include <stdio.h>
#include <math.h>
int getresult(int arr[], int n)
{
     int min = 0, max = 0;
     /*If there is only one element then return it as min and max both*/
     if (n == 1)
          min = max = arr[0];

     /* If there are more than one elements, then initialize min and max*/
     if (arr[0] > arr[1])
     {
          max = arr[0];
          min = arr[1];
     }
     else
     {
          max = arr[1];
          min = arr[0];
     }
     for (int i = 2; i < n; i++)
     {
          if (arr[i] > max)
               max = arr[i];
          else if (arr[i] < min)
               min = arr[i];
     }
     printf(" Minimum element: %d", min);
     printf(" Maximum element: %d", max);
}

int main()
{
     int arr[] = {200, 191, 112, -11, 330, 60};
     int n = 6;
     getresult(arr, n);
}
```

## *Strings*

### Declaring and initializing a string variables

```
char array[][20] = {“name”, “lastName”, “Job”};
```

### String Input and Output

```
#include <stdio.h>
int main()
{
     char str[20];
     printf("Name: ");
     scanf("%[^\n]", &str);
     printf("%s", str);

     char text[20];
     printf("\nPass text:\n");
     gets(text);
     printf("%s", text);

     int size;
     printf("\nEnter size: ");
     scanf("%d", &size);
     char array[size][20];
     printf("\nEnter list of names:\n");
     for (int i = 0; i < size; i++)
          scanf("%s\n", &array[i]);

     printf("\n-----------------\n");
     for (int i = 0; i < size; i++)
          printf("%s\n", array[i]);

     return 0;
}
```

### String Handling Functions

| Method | Description |
|---|---|
| Strcat() | It is used to concatenate(combine) two strings |

```
#include <stdio.h>
#include <string.h>
int main()
{
     char str1[] = "hello";
     char str2[] = " world";
     char str3[] = " to new world";
     strcat(str1, str2);
     strcat(str1, str3);
     puts(str1);

     char name[] = "Muhammed laoukili better";
     int size = strlen(name);
     printf("size = %d", size);

     return 0;
}
```

### Strlen()    It is used to show the length of a string

```c
#include <stdio.h>
#include <string.h>
int main()
{
    char name[] = "My name is Muhammed";
    int size = strlen(name);
    printf("Size of the string = %d", size);

    return 0;
}
```

### Strrev()    It is used to show the reverse of a string

```c
#include <stdio.h>
#include <string.h>
int main()
{
    char word[] = "live";
    strrev(word);
    printf("live = %s", word);

    return 0;
}
```

### Strcpy()    Copies one string into another

```c
#include <stdio.h>
#include <string.h>
int main()
{
    char word[] = "live";
    strcpy(word, "love");
    printf("live = %s", word);

    return 0;
}
```

### Strcmp()    It is used to compare two string

```c
#include <stdio.h>
#include <string.h>
int main()
{
    char word[] = "live";
    if (strcmp(word, "live") == -1)
        printf("Words not the same");
    else
        printf("words are the same");

    return 0;
}
```

- *Pointers*

### *what are the pointers?*

A “variable-like” reference that holds a memory address to another variable, array, etc.

Some tasks are performed more easily with pointers

### *Advantages of pointers:*

- ✓ Less time in program execution
- ✓ Working on the original variable
- ✓ With the help of pointers, we can create data structures such as (linked list, stacks, queues...)
- ✓ Returning more than one value from function
- ✓ Searching and sorting large data very easily
- ✓ Dynamically memory allocation

### *Examples:*

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

void swap(int *a, int *b)
{
     int temp = *a;
     *a = *b;
     *b = temp;
}

int main()
{
     int array[] = {13, 14, 35, 46, 57, 68, 79, 90, 91};
     int *p;
     p = array;

     *p + 3;   // 16
     *(p + 3); // 46

     &p + 2; // The address of pointer behind the pointer p (rarely used)

     &array[2];

     &array[5] - 4; // the address of the component array [1]
     &array[1];

     array + 4; // the address of the component array[4]
     &array[4];

     &array[8] - p; // 8 { value (index 8) }

     p + (*p - 10); // the address at the component array[3]
     &array[3];
     array[3];

     *(p + *(p + 8) - array[7]); // ~~24~~  14
     /*
     *(p + *(p + 8) - array[7]) = *(p + *(p + 8) - 90)
                                   = *(p + 91 - 90)
                                   = *(p + 1)
                                   = ~~24~~ 14
     */

     *p; // 13

     // swap using pointers
     int a = 4, b = 8;
     swap(&a, &b);
     printf("a = %d\t|\tb = %d", a, b);

     return 0;
}
```

- *Functions*
    - Function parameters
    - Function declaration
    - Recursion

```c
#include <stdio.h>
int sum(int k);
int main()
{
	int result = sum(10);
	printf("%d", result);
	return 0;
}
int sum(int k)
{
	if (k > 0)
		return k + sum(k - 1);
	else
		return 0;
}
```

- *Math functions*

```c
#include <math.h>
#include <stdio.h>
int main()
{
	printf("%f", sqrt(16));
	printf("%f", ceil(1.4));
	printf("%f", floor(1.4));
	printf("%f", pow(4, 3));
	return 0;
}
```

| Function | Description |
|---|---|
| abs(x) | Returns the absolute value of x |
| acos(x) | Returns the arccosine of x |
| asin(x) | Returns the arcsine of x |
| atan(x) | Returns the arctangent of x |
| cbrt(x) | Returns the cube root of x |
| cos(x) | Returns the cosine of x |
| exp(x) | Returns the value of Ex |
| sin(x) | Returns the sine of x (x is in radians) |
| tan(x) | Returns the tangent of an angle |

- *Structures*

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

typedef struct classe
{
     char name[20];
     int mark;
} classe;

int main()
{
     int i, n_students;
     printf("Students = ");
     scanf("%d", &n_students);
     classe student[n_students];

     for (i = 0; i < n_students; i++)
     {
          printf("Enter name for student %d: ", i + 1);
          scanf("%s", student[i].name);
          printf("Enter mark for student %d: ", i + 1);
          scanf("%d", &student[i].mark);
     }
     printf("\n\n");
     printf("Name\t\t|\t\tNote\t\t|Marks\n---------------------------------------------------------\n");
     for (int j = 0; j < n_students; j++)
     {
          if (student[j].mark > 12)
          {
               printf("%d|| %s\t|\t Greate you are a legend!!\t|\t%d\n", j + 1, student[j].name, student[j].mark);
               continue;
          }
          if (student[j].mark < 10)
               printf("%d|| %s\t|\t You are going to fail!!!\t|\t%d\n", j + 1, student[j].name, student[j].mark);
          else
               printf("%d|| %s\t|\t Do more effort next time\t|\t%d\n", j + 1, student[j].name, student[j].mark);
     }

     return 0;
}
```

**Created by Muhammed ElIdrissi Laoukili**

لا اله الا الله