C programming | part 1

Note: These notes are not meant to cover everything on C! Please read the materials provided on course website!

Types

integer	int	16 bits	[-32768, 32767]
long	long	32 bits	[-2147483648, 2147483647]
character	char	8 bits	
float	float	32 bits	
double	double	64 bits	

Declaration of Structure

```
struct  NAME {
        variable declarations
        ...
};
```

Called members or fields of the structure!

Example ⇒

```
struct BANK_ACCOUNT {
    Char  name[32];
    int   id;
    float balance;
};
```

Declaration of variables

```
int i; //variable i of type int
int i,j,k;
int v=0; //initialized

float my-cool_variable;

struct BANK-ACCOUNT ba; //variable ba of type BANK_ACCOUNT
BANK-ACCOUNT anExampleOfBankAccount;
```

```
int rect[10];
char my_name[32];
double mat[10][20];
```

← A char requires 8 bits = 1byte
my_name will have allocated for it 32 * 1 bytes!

#define directive

```
#define ROWS 10
#define COLS 20

int matA[ROWS][COLS];
int a=ROWS;
```

format specifiers

format specifier
↓

```
char s = 'A';
printf ("The character variable s has value %c \n", s);
```

%c	char
%d	int
%f	float, double
%s	string of characters

Conditional Operator

```
E1 ? E2 : E2
```

expressions

```
float a=10, b=7, c=-1;
c = a>b ? 22 : c;
char answer;
answer = a==b ? 'A' : 'B';
```

instructions

1 if

```
if (expression) {
      ...
} else {
      ...
}
```

Example:

```
float x, y, a;
x = 3;
y = 4; a = 5;
if (x < 0) {
   y = a + 5;
} else {
   y = a - 5;
}
```

Note: we could also do:

```
y = (x < 0) ? (a+5) : (a-5);
```

2 while

```
while (expression) {
      ...

}
```

Example:

```
while (x > 0) {
     printf(" y is %f ", y);
     x = x - 1;
}
```

3 for

```
for ( E1; E2; E3) {
        ...
}
     expressions
```

Example:

```
long i = 0;
for (i=0; i<10; i++) {
     printf("Say hi %d", i);
}
```

4 do-while

```
do {


} while (expression);
```

Exemple:

```
do {
   printf("Say hi %d", i);
   i++
} while (i < 10);
```

⑤ continue;

It can be used inside instructions for, while, do-while
It stops execution of current iteration and continues to the next.

Exomple:

```
int i, a,b;
for (i=0; i<10; i++) {
    a = i;
    if (i==5) {
        continue;
    }
    b= i+10;
}
```

⑥ break;

It terminates the instruction for, while, do-while

Exemple:

```
int i=10, a, b;
while (i>=0) {
    a = i
    b= a+1;
    if (b==3) {
        breek;
    }
}
```

⑦ switch

```
switch(E1) {
case c1:
        ...
        breek;
case c2:
        ...
        breek;
 :
case cn:
        ...
        breek;
default:
        ...
}
```

Exemple:

```
int i=0;
int a= i+3;
switch (a) {
    case 2:
        puts ("Mondey"); breek;
    case 3:
        puts ("Ooops"); breek;
    default:
        puts("Are you kidding me ?");
}
```

Variable initialization

```
int  n=100;
#define MAX  100
int i = MAX *2;
char  c = 'a';
int  tab[10] = { 0, 1, 2, 3, 4, 9, 8, 7, 6, 5 };
                 ↑              ↑           ↑
               tab[0]         tab[5]     tab[10]

double  d[20] = { 0, 1, -1, 3, -2 };
char  er[] = { 'e', 'r', 'r', 'o', 'r', '\0' };
                                          ↑
                                  this is the NULL character
                                  which terminates any
                                  array of characters!
char  er[] = "error";

int  A [2][3] = {
         {-1, 0},
         {-1, 2},
         {3, 7}
};

 float [][2] = {
        {1.5, 2},
        {3.1, 4.2}
 }
```

Note: It's good programming practice/style to initialize all your (simple) variables!

Example 1: find maximum value in an array of positive numbers

```
int main () {
    int tab[10] = { 4, 3, 2, 10, 11, 12, 9, 8, 7, 0};
    int i=0;
    int max = -1;

    for (i=0; i<10; i++){
        if (max < tab[i])   max = tab[i];
    }
    printf ("Maximum value is %d", max);
}
```

Example 2 Bubble sort (Assignment: ShellSort, QuickSort)

```
int main () {
    int tab [6] = { 5, 4, 3, 1, 2, 6 };
    int n=6; int temp;
    int swapped = 0; //will use it as a Boolean variable.
    do {
        swapped = 0; //means FALSE
        for (i=1; i<=n-1; i++) {
            if (tab[i-1] > tab[i]) {
                temp = tab [i-1];
                tab [i-1] = tab [i]
                tab[i] = temp;
                swapped = 1; //means TRUE
            }
        }
        n=n-1
    }while (swapped);
    printf("Maximum value is %d", tab[5]);
}
```