

Programmation

- if ()
{
else
}
;
- printf ("%c", a);
- scanf ("%c", &a);
- & & : and
|| : or
! : Not
- switch (expression)
{
case 1 : ---- ;
break;
:
default : ---- ;
break;
}

- while ()
{
}
;
- do {
} while (expression);
- for (i=0; i<N; i++)
{
}
}

- Taille du Tableau
int Tab[Nmax];
int N;
Do { printf("Donner la taille du Tab: ?");
scanf ("%d", &N);
if ((N<0) || (N>Nmax))
printf ("%d est invalide")
} while ((N<0) || (N>Nmax));

- remplissage du Tab :

```
for (i = 0; i < N; i++)
{ printf("Donner Tab [%d] = ?", i);
scanf("%d", &Tab[i]); }
```
- affichage du Tab :

```
for (i = 0; i < N; i++)
{ printf("Tab [%d] = %d\n", i, Tab[i]); }
```
- lorsque on utilise de fonction math (PI...) => include <Math.h>
#
- saisie d'une chaîne :
gets(ch);
- affiche d'une chaîne :
puts(ch);
- longueur du chaîne :
l = strlen(ch);
- comparer 2 chaînes :
strcmp(ch1, ch2); (donne 0 si =
1 si ch1 > ch2
-1 si ch1 < ch2)
- s'il est minuscule :
islower(ch[i])

- renché le caractère maj :
ch[i] = toupper(ch[i]);
- lorsque on utilise le caractère, il faut ajouter
#include <string.h>
#include <ctype.h>
- Tester s'il est un espace :
isspace(ch[i])
- for (i = 0, j = K-1; i < j; i++, j--)
pour cours de début et de fin
- structure :
define NP
struct point
{ char c;
int x, y; };
main()
{ struct point p;
printf("donner nom du pt");
scanf("%c", &p.c);
printf("donner les coordonnées");
scanf("%d%d", &p.x, &p.y);
printf("point %c de coord %d%d", p.c, p.x, p.y); } (f(venir))

- Pointer is a special variable that is capable of storing some address
- it points to a memory location where the first byte is stored.
- `strcpy(ch1, ch2)` \Rightarrow copie `ch2` vers `ch1`
- `strcat(ch1, ch2)` \Rightarrow ajout de `ch2` à la fin de `ch1`
- `strncpy(ch1, ch2, n)` \Rightarrow copie `n` caractères de `ch2` vers `ch1`
- `strncat(ch1, ch2, n)` \Rightarrow ajout de `n` caractères de `ch2` vers `ch1`.

Tri de Tableaux

```
for (i = 0; i < N - 1; i++)
{
    for (j = i + 1; j < N; j++)
        if (Tab[j] < Tab[i])
        {
            val = Tab[j];
            Tab[j] = Tab[i];
            Tab[i] = val;
        }
}
```

```
printf("préciser la valeur à ajouter au tab: ");
scanf("%d", &val);
```

- recherche de val:


```
for (i = 0; i < N) {
                if (Tab[i] != val) { i++; }
                if (Tab[i] == val) {
                    pos = i + 1;
                    Break;
                }
            }
```
- `printf("%p", &a)`: affiche l'adresse
- Un pointeur est une variable qui peut contenir l'adresse d'une autre variable.
- `&` : l'adresse d'une variable
- `*` : contenu d'une adresse
- ```
int *p; int *p = &a;
p = &a;
```
- Le nom d'un Tableau représente l'adresse de son premier élément.
 

```
&int *p;
p = T; // premier élément
```
- `T` : l'adresse de `T[0]`  
`T + i` : l'adresse de `T[i]`  
`* (T + i)` : le contenu de `T[i]`

- moyenne du Tab :

```
double MoyenneTab (int Tab[], int N)
```

```
{ int i;
```

```
 double moy;
```

```
 if (N)
```

```
 { for (i=0, Moy=0; i<N; i++)
```

```
 Moy += Tab[i];
```

```
 return Moy/N;
```

```
 else
```

```
 return N;
```

```
}
```