

Série d'exercices 0x11

Structures itératives

INFO1-TIN-1

23 mars 2020

Problème 1 – Affichage de nombres entiers

- (i) Écrire une fonction qui affiche les nombres entiers de 1 à 100 en employant une boucle `for`.
- (ii) Réécrire ce même programme avec une boucle `while`.
- (iii) Réécrire ce même programme avec une boucle `do`.
- (iv) Quelle est la structure de contrôle la plus adaptée à cette application ?

Problème 2 – Utilisation particulière de for

Expliquez quelle est la fonctionnalité du programme ci-dessous :

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

int main(void)
{
    for (size_t i = 0; i * i < 1000; i++,
        printf("\n"))
        printf("%d", i);
}
```

Problème 3 – Comptage

- (i) Écrire un programme qui lit une ligne de texte saisie par l'utilisateur et qui affiche le nombre de caractères saisis.
Indication : le dernier caractère saisi par l'utilisateur est toujours le caractère `LF`.
- (ii) Écrire une fonction qui lit une ligne de texte saisi par l'utilisateur tant qu'il n'y a que des 0 et des 1, et qui retourne l'entier correspondant au code binaire saisi.
- (iii) Écrire une fonction qui reçoit un nombre entier 16-bits en paramètre, et qui affiche sa représentation en binaire à l'écran.

Problème 4 – Testez vos connaissances

Remarque : ces exercices comportent des instructions qui ne constituent pas des exemples à suivre. Ils ont

pour but de vous permettre d'analyser des formes tortueuses des instructions pour affiner votre compréhension.

Pendant les labos, veillez à écrire du code aussi limpide et bien structuré que possible.

4.1 do...while

Indiquer si les affirmations suivantes sont justes ou fausses. Dans les cas où elles sont fausses, expliquer ce qui serait correct. Dans une instruction `do...while` :

- (i) Les instructions de la boucle sont toujours exécutées au moins une fois.
- (ii) Comme un mot réservé spécifique commence et termine la boucle, on n'a pas besoin de créer un bloc lorsque l'on a plusieurs instructions.
- (iii) La condition se trouvant en fin de boucle, on sort de la boucle lorsque la condition est vraie.
- (iv) Le type de la condition peut être `char`.
- (v) Les instructions de la boucle ne peuvent pas être une autre boucle `do...while`.

4.2 while

Indiquer pour chaque groupe d'instruction ci-dessous ce qui sera affiché à l'exécution.

(i)

```
int i = 0;
while ( i - 10 )
{ i += 2; printf ( "%i\n", i );
}
```

(ii)

```
int i = 0;
while ( i - 10 )
i += 2; printf ( "%i\n", i );
```

(iii)

```
int i = 0;
while ( i < 11 )
{ i += 2; printf ( "%i\n", i );
}
```

(iv)

```
int i = 11;
while ( i-- )
{ printf ( "%i\n", i-- );
}
```

(v)

```
int i = 12;
while (i--)
{ printf ("%i\n", --i);
}
```

(vi)

```
int i = 0;
while ( i++ < 10 )
{ printf ( "%i\n", i-- );
}
```

(vii)

```
i = 1;
while ( i <= 5 )
{ printf ( "%i\n", 2 * i++ );
}
```

(viii)

```
int i = 1;
while ( i != 9 )
{ printf ( "%i\n", i = i + 2 );
}
```

(ix)

```
int i = 1;
while ( i < 9 )
{ printf ( "%i\n", i += 2 ); break;
}
```

(x)

```
int i = 0;
while ( i < 10 )
{ continue;
  printf ( "%i\n", i += 2 );
}
```

4.3 for

Indiquer pour chaque groupe d'instruction ci-dessous ce qui sera affiché à l'exécution.

(i)

```
for (int i = 'a'; i < 'd'; printf ("%i\n", ++i));
```

(ii)

```
for (int i = 'a'; i < 'd'; printf ("%c\n", ++i));
```

(iii)

```
for (int i = 'a'; i++ < 'd'; printf ("%c\n", i ));
```

(iv)

```
for (int i = 'a'; i <= 'a' + 25; printf ("%c\n", i++ ));
```

(v)

```
for (int i = 1 / 3; i ; printf("%i\n", i ++ ));
```

(vi)

```
for (int i = 0; i != 1 ; printf("%i\n", i += 1 / 3 ));
```

(vii)

```
for (int i = 12, k = 1; k++ < 5 ; printf("%i\n", i-- ));
```

(viii)

```
for (int i = 12, k = 1; k++ < 5 ; k++, printf("%i\n", i-- ));
```

(ix)

```
for (int i = 1 ; printf ("Resultat: ") , (i+=1)<5 ; printf("%i\n",i));
```