

Q 01: Que signifie GCC ?

GCC est un ensemble de compilateurs libres créé par le projet GNU.
Il sert à transformer le code source que tu écris en langage machine que l'ordinateur peut exécuter.

Q 02: OK, alors que signifie GNU ?

GNU est un acronyme récursif qui signifie: GNU n'est pas Unix.

Q 03: C'est quoi le problème avec GCC, au fait ?

- + Contrairement à certains IDE modernes, il n'a pas d'interface graphique native, ce qui peut décourager les nouveaux développeurs
- + Les messages d'erreur peuvent être très verbeux et techniques.

Q 04: Comment GCC est-il bon pour trouver des erreurs dans les programmes ?

GCC peut détecter et signaler les erreurs dans les codes avant même que le programme ne s'exécute.

Q 05: Pourquoi C est-il si laconique ? J'ai l'impression que les programmes seraient plus lisibles si C utilisait `begin` et `end` au lieu de `{` et `}`, `integer` au lieu de `int`, et ainsi de suite.

Pour gagner du temps et économiser de la place.

Q 06: Dans certains livres sur le C, la fonction `main` se termine par `exit(0)` au lieu de `return 0`. Est-ce la même chose ?

Non, car `exit(0)` termine tous le code et quit le programme, mais `return 0` quit la fonction `main` seulement ou autre fonction s'elle dans autre fonction.

Q 07: Que se passe-t-il si un programme atteint la fin de la fonction `main` sans exécuter un `return` ?

Le programme excuse normalement.

Q 08: Le compilateur supprime-t-il un commentaire entièrement ou le remplace-t-il par un espace blanc ?

Le compilateur ne lire pas le commentaire et par la suite il le supprime entièrement.

Q 09: Comment puis-je savoir si mon programme a un commentaire non terminé ?

Pour écrire un commentaire soit en utilise `//` ou bien `/**/` avec `/*` est l'ouverture de lui et `*/` est la fermeture. Alors sont `*/` le commentaire non terminé.

Q 10: Est-il légal d'imbriquer un commentaire à l'intérieur d'un autre ?

On peut.

Q 11: D'où vient le nom du type float ?

Ce terme devient de virgule flottante.

Q 12: Pourquoi les constantes à virgule flottante doivent-elles se terminer par la lettre f ?

1.1) Non

1.2) a) les directives : #include<stdio.h> et les instructions : printf("Parkinson's Law:\nWork expands so as to "); printf("fill the time\n"); printf("available for its completion.\n"); return 0;

**b) The output {Parkinson's Law:
Work expands so as to fill the time
available for its completion.
}**

1.4)

```
Start here × main.c ×
1  /* Computes the dimensional weight of a 12"x10"x8" box */
2  #include <stdio.h>
3
4  int main(void) {
5      int height, length, width, volume;
6
7      height = 8;
8      length = 12;
9      width = 10;
10
11      volume = height * length * width;
12
13
14      printf("Dimensions: %dx%dx%d\n", length, width, height);
15      printf("Volume (cubic inches): %d\n", volume);
16      printf("Dimensional weight (pounds): %d\n", (volume + 165) / 166);
17      return 0;
18  }
19
```

1.4)

```
main.c - Code::Blocks 25.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global> main(void) : int
C:\Users\Youcode\Desktop\SAS\main.exe
The X value : 0
The Y value : 0
The M value : 0.000000
The N value : 138842393374684553194552778817536.000000
Process returned 0 (0x0) execution time : 1.028 s
Press any key to continue.
File L... Message
=== Build file: "no target" in "no project" (compiler: unkn...
=== Build finished: 0 error(s), 0 warning(s) (0 minute(s), ...
C:\Users\Youcode\Desktop\SAS\main.c C/C++ Windows (CR+LF) WINDOWS-1252 Line 10, Col 37, Pos 278 Insert Read/Write default
```

```
main.c - Code::Blocks 25.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global> The X value : 0
The Y value : 0
The M value : 540749463552.000000
The N value : 0.000000
Process returned 0 (0x0) execution time : 1.023 s
Press any key to continue.
File L... Message
=== Build file: "no target" in "no project" (compiler: unkn...
=== Build finished: 0 error(s), 0 warning(s) (0 minute(s), ...
C:\Users\Youcode\Desktop\SAS\main.c C/C++ Windows (CR+LF) WINDOWS-1252 Line 10, Col 37, Pos 278 Insert Read/Write default
Code::Blocks Search results Cccc Build log Build messages CppCheck/Vera++ CppCheck/Vera++ messages Cscope Debugger Di...
```

1.7)

5) (a)

6)

7)(a)(c)(e)

1.8)

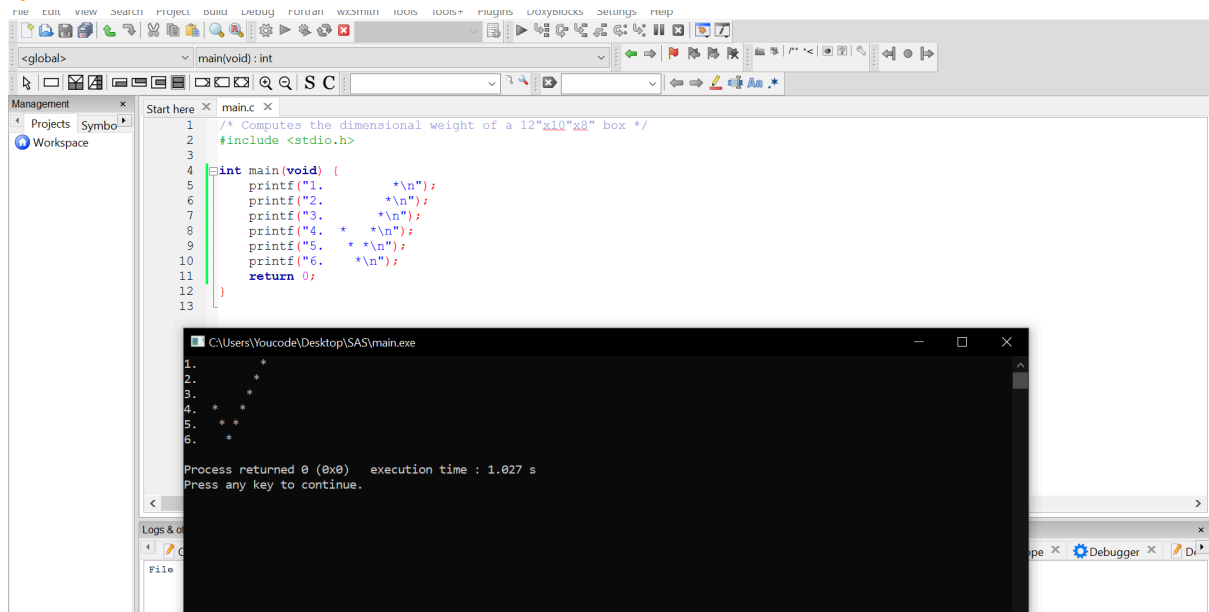
8) 11 jetons

9) $\text{answer} = (3 \cdot q - p \cdot p) / 3;$

10) L'espace entre les jetons de ces ligne : `"printf("Dimensions: %dx%dx%d\n", length, width, height); printf("Volume (cubic inches): %d\n", volume); printf("Dimensional weight (pounds): %d\n", weight); "`

CHAPITRE IV

1)



The screenshot shows a C program in a debugger. The program is named `main.c` and is located at `C:\Users\Youcode\Desktop\SAS\main.exe`. The code is as follows:

```
1 /* Computes the dimensional weight of a 12"x10"x8" box */
2 #include <stdio.h>
3
4 int main(void) {
5     printf("1.      *\n");
6     printf("2.      *\n");
7     printf("3.      *\n");
8     printf("4.  *  *\n");
9     printf("5.  *  *\n");
10    printf("6.  *  *\n");
11    return 0;
12 }
13
```

The program is executed, and the output is displayed in a console window. The output shows a box of stars, with the first three lines having one star each, and the next three lines having two stars each. The console window also displays the message "Process returned 0 (0x0) execution time : 1.027 s" and "Press any key to continue."

2)

The screenshot shows the Code::Blocks IDE with a C program open in the editor. The program calculates the volume of a cylinder. The output window shows the result of the execution.

```
1 /* Computes the dimensional weight of a 12"x10"x2" box */
2 #include <stdio.h>
3 #include <math.h>
4
5 int main(void) {
6     float r = 10;
7     float pi = 3.14;
8
9     float v = (4.0/3.0) * pi * r * r * r;
10    printf("The Volume is : %f", v);
11    return 0;
12 }
13
```

Execution Output:

```
C:\Users\Youcode\Desktop\SAS\main.exe
The Volume is : 4186.666992
Process returned 0 (0x0)   execution time : 1.041 s
Press any key to continue.
```

The screenshot shows the Code::Blocks IDE with a C program open in the editor. The program calculates the volume of a cylinder. The output window shows the result of the execution.

```
1 /* Computes the dimensional weight of a 12"x10"x2" box */
2 #include <stdio.h>
3 #include <math.h>
4
5 int main(void) {
6     float r = 10;
7     float pi = 3.14;
8
9     float v = (4/3) * pi * r * r * r;
10    printf("The Volume is : %f", v);
11    return 0;
12 }
13
```

Execution Output:

```
C:\Users\Youcode\Desktop\SAS\main.exe
The Volume is : 3140.000000
Process returned 0 (0x0)   execution time : 1.023 s
Press any key to continue.
```

3)

The screenshot shows a C program in a code editor and its execution in a terminal window. The code defines a function `main` that prompts the user for a radius `r`, reads it, and calculates the volume `v` of a cylinder with a fixed height of 12 using the formula $v = \frac{4}{3} \pi r^2$. The terminal shows the user inputting 43, resulting in a volume of 249652.000000.

```
1  /* Computes the dimensional weight of a 12"x10"x8" box */
2  #include <stdio.h>
3  #include <math.h>
4
5  int main(void) {
6      float r;
7      printf("Entrer un volum : \n");
8      scanf("%f",&r);
9      float pi = 3.14;
10
11     float v = (4/3) * pi * r *r *r;
12     printf("The Volume is : %f", v);
13     return 0;
14 }
15
```

Entrer un volum :
43
The Volume is : 249652.000000
Process returned 0 (0x0) execution time : 5.076 s
Press any key to continue.

4)

The screenshot shows a C program in a code editor and its execution in a terminal window. The code defines a function `main` that prompts the user for a dollar amount `d`, reads it, calculates a 5% tax, and prints the final amount in cents (`c`) and dollars (`d`). The terminal shows the user inputting 111, resulting in 116.550003 cents or 116.550000 dollars.

```
1  /* Computes the dimensional weight of a 12"x10"x8" box */
2  #include <stdio.h>
3  #include <math.h>
4
5  int main(void) {
6      float d,c;
7      printf("Entrer un montant en Dollar($) : ");
8      scanf("%f",&d);
9      float taxe = d / 100 * 5;
10     float finalMontant = d + taxe;
11     c = finalMontant * 100;
12     printf("Votre montant avec taxes est : %f($$) ou bien %f cent", finalMontant, c);
13     return 0;
14 }
15
```

Entrer un montant en Dollar(\$) : 111
Votre montant avec taxes est : 116.550003(\$\$) ou bien 11655.000000 cent
Process returned 0 (0x0) execution time : 4.104 s
Press any key to continue.

5)

The screenshot shows a C program in a code editor. The program defines a function `main` that prompts the user for a value `x`, calculates a polynomial equation, and prints the result. The terminal window shows the program being executed with `x = 4`, resulting in a value of `3270.000000`.

```
//3x5+2x4-5x3-x2+7x-6
#include <stdio.h>
#include <math.h>

int main(void) {
    float x;
    printf("Entrez une valeur pour le variable 'x' : ");
    scanf("%f", &x);
    float equation = (3* x*x*x*x*x) + (2* x*x*x*x) - (5* x*x*x) - (x*x) + (7 * x) - 6;
    printf("La solution de l'équation basons sur votre valeur est : %f", equation);

    return 0;
}
```

```
C:\Users\Youcode\Desktop\SAS\main.exe
Entrez une valeur pour le variable 'x' : 4
La solution de l'équation basons sur votre valeur est : 3270.000000
Process returned 0 (0x0)   execution time : 4.282 s
Press any key to continue.
```

6)

The screenshot shows a C program in a code editor. The program defines a function `main` that prompts the user for a value `x`, calculates a polynomial equation, and prints the result. The terminal window shows the program being executed with `x = 13`, resulting in a value of `1159932.000000`.

```
//3x5+2x4-5x3-x2+7x-6
#include <stdio.h>
#include <math.h>

int main(void) {
    float x;
    printf("Entrez une valeur pour le variable 'x' : ");
    scanf("%f", &x);
    float equation = (((3*x+2) * x - 5) * x - 1)*x + 7) * x - 6);
    printf("La solution de l'équation basons sur votre valeur est : %f", equation);

    return 0;
}
```

```
C:\Users\Youcode\Desktop\SAS\main.exe
Entrez une valeur pour le variable 'x' : 13
La solution de l'équation basons sur votre valeur est : 1159932.000000
Process returned 0 (0x0)   execution time : 27.594 s
Press any key to continue.
```

7)

The screenshot shows a C program in a code editor and its execution in a terminal window. The code calculates the number of 20, 10, 5, and 1 dollar bills for a given amount. The terminal shows the input 132 and the corresponding output for each bill denomination.

```
main(void) : int
3 #include <math.h>
4
5 int main(void) {
6     int montant;
7     printf("Entrez un montant : ");
8     scanf("%d", &montant);
9     int twentyDollarDiv = montant % 20;
10    int tenDollarDiv = twentyDollarDiv % 10;
11    int fiveDollarDiv = tenDollarDiv % 5;
12    int oneDollarDiv = fiveDollarDiv % 1;
13    int billets_twenty = (montant - twentyDollarDiv) / 20;
14    int billets_ten = (twentyDollarDiv - tenDollarDiv) / 10;
15    int billets_five = (tenDollarDiv - fiveDollarDiv) / 5;
16    int billets_one = (fiveDollarDiv - oneDollarDiv);
17
18    printf("Billets de 20$ : %d\n", billets_twenty);
19    printf("Billets de 10$ : %d\n", billets_ten);
20    printf("Billets de 5$ : %d\n", billets_five);
21    printf("Billets de 1$ : %d\n", billets_one);
22
23    return 0;
24
25
26
```

Process returned 0 (0x0) execution time : 4.174 s

8)

The screenshot shows a C program in a code editor and its execution in a terminal window. The code calculates the remaining balance of a loan after a series of payments, considering interest. The terminal shows the input values and the resulting balance after each payment.

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int main(void) {
5     float x, paiement, tauxM;
6     printf("Entrez le montant du prêt : ");
7     scanf("%f", &x);
8     printf("Entrez le taux d'intérêt : ");
9     scanf("%f", &tauxM);
10    printf("Entrez le paiement mensuel : ");
11    scanf("%f", &paiement);
12    float taux = tauxM / 100 / 12;
13    float resultat = (x - paiement) + (x * taux);
14    float resultatDeux = (resultat - paiement) + (resultat * taux);
15    float resultatTrois = (resultatDeux - paiement) + (resultatDeux * taux);
16    printf("Solde restant après le premier paiement : %f\n", resultat);
17    printf("Solde restant après le premier paiement : %f\n", resultatDeux);
18    printf("Solde restant après le premier paiement : %f\n", resultatTrois);
19
20    return 0;
21
22
```

Process returned 0 (0x0) execution time : 20.155 s