# Partie I

*QCM : Pour chacune des questions ci-dessous, indiquez, sur la feuille de réponse, qui vous glisserez dans votre copie, la bonne réponse de manière claire et lisible (réponses illisibles seront annulées).*

*Points : 3*

1. L’expression « e \*= e ; » :
2. contient une erreur de syntaxe
3. équivaut à l’expression « e = e \* 2 ; »
4. calcule le carré de e (*e2*)
5. aucune des réponses précédentes n’est correcte
6. Que fait l’expression : « if (e%2 != 0) p\*=y; » ?
7. Rien, elle contient une erreur de syntaxe
8. Elle multiplie la valeur de p par celle de y
9. Elle multiplie la valeur de p par celle de y uniquement si e est impair
10. Elle multiplie la valeur de p par celle de y uniquement si e est pair
11. Aucune des réponses précédentes n’est correcte
12. Le programme

int main() {

int i=0 ; while(i<10) printf("%d ",i); i++;

}

1. va afficher 9 nombres
2. va afficher 11 nombres
3. va afficher 10 nombres
4. va boucler indéfiniment
5. contient une erreur de syntaxe
6. Aucune des réponses précédentes n’est correcte
7. Les instructions suivantes « if (h>5 && h<=18) printf("Bonjour") ;   
   else printf ("Bonsoir") ; } »
8. affichent toujours « Bonjour »
9. affichent toujours « Bonsoir »
10. affichent « Bonjour » si la valeur de h est égale à 5
11. affichent « Bonsoir » uniquement si h est inférieur à 5 ou si h est supérieur ou égale à 18
12. Aucune des réponses précédentes n’est correcte
13. L’instruction « if (a%2 == 0) printf("bonjour") ; »
14. contient une erreur de syntaxe
15. n’affiche rien, quelque soit la valeur de a
16. affiche « bonjour » quand a est un entier impair
17. affiche « bonjour » quand a est un entier pair
18. Aucune des réponses précédentes n’est correcte
19. L’opérateur == permet en langage C de :
20. réaliser une affectation
21. tester une égalité
22. comparer deux variables
23. convertir un int en float
24. Aucune des réponses précédentes n’est correcte
25. Le programme « int main(){ int b=5, x=12, z ; z = (2+5\*x+4)/b-3 ; printf ("%d",z); } » affiche la valeur :
26. 10
27. 10.2
28. 17
29. 17.6
30. Aucune des réponses précédentes n’est correcte
31. Quelle sera la valeur de j après l’exécution du bloc d’instructions suivant :

j = 0;

switch (i) {

case 3:

j++;

case 2:

j+=2;

case 1:

j+=3;

}

1. 3 si i=3, 2 si i=2, 1 si i=1
2. 1 si i=3, 2 si i=2, 1 si i=3
3. 6 si i=3, 5 si i=2, 3 si i=1
4. 6 quelque soit la valeur de i
5. Aucune des réponses précédentes n’est correcte
6. L’instruction «  if (size=0) i++ ; else i--; »
7. contient une erreur
8. incrémente la variable i si size est positif
9. décrémente la variable i si size est négatif
10. incrémente la variable i si size est différent de 0
11. décrémente la variable i si size est différent de 0
12. Le programme

int main () {

int ASCII;

ascii = getchar() ;

printf ("%d\n", ascii);

}

1. contient une erreur
2. va afficher le code ASCII du caractère fourni par l’utilisateur
3. va afficher le caractère fourni par l’utilisateur
4. Aucune des réponses précédentes n’est correcte

# Partie II

*Questions à lacunes : recopiez sur votre copie, de manière claire et lisible, le n° de la ligne (indiqué à gauche de la ligne de code) suivi de toute la ligne avec les lacunes remplies dans l’ordre. Recopiez uniquement les lignes contenant les lacunes. Les copies illisibles seront annulées.*

*Points : 2*

1. Remplissez les lacunes soulignées dans le programme ci-dessous :

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | #include <stdio.h>  int main ()  {  float x,y,t;  printf ("Entrez x : ");  \_\_\_\_\_\_ ("\_\_\_", &x);  printf ("Entrez y : ");  \_\_\_\_\_\_ ("\_\_\_", &y);  if (x<y) {  t = \_\_\_\_;  x = y;  y = \_\_\_\_;  }  printf ("Ecart : %f", (x-y));  } |

# Partie III

*Traces : Réalisez la trace d’exécution des programmes indiqués sur votre copie. Ecrivez de manière claire et lisible selon le modèle ci-dessous (questions illisibles seront annulées).*

|  |  |  |  |
| --- | --- | --- | --- |
| *(Question 12) Pont d’observation* | *Variable* | *Variable* | *…* |
| *Pont d’observation 1* |  |  |  |
| *…* |  |  |  |

*Points : 5*

1. Tracez l’exécution du programme suivant.

|  |
| --- |
| #include<stdio.h>  main(){  int min, max;  int reste;  int i, n;  min = 2;  max = 5;  printf ("Les premiers entre %d et %d sont : ", min, max);  n = min;  //pont d'observation 1  while (n<=max) {  i=2;  reste = 1;  //pont d'observation 2  do {  if (n != i)  reste = n%i;  i++;  //pont d'observation 3  } while (i<=n && reste != 0);  //pont d'observation 4  if (reste != 0) {  printf (" %d ", n);  }  n += 1; //pont d'observation 5  }  //pont d'observation 6  } |

# Partie IV

*Pour chacune des questions ci-dessous, écrivez le programme pour le problème indiqué, suivant les indications données. Ecrivez de manière claire et lisible (questions illisibles ne seront pas prises en considération).*

*Points : 10 (question 13 : 6 points, question 14 : 4 points)*

1. Un nombre est parfait quand il est égal à la somme de ses diviseurs. Par exemple :

6 = 1 + 2 + 3 est un nombre parfait.

28 = 1 + 2 + 4 + 7 + 14 est aussi un nombre parfait.

8 ≠ 1 + 2 + 4 n’est pas un nombre parfait.

*Rappel : les diviseurs d’un nombre entier positif n sont de tous les nombres entiers compris entre 1 et n-1 qui divisent n.*

Ecrire un programme qui calcule et affiche si les nombres parfait entre 5 et 100.

1. La compagnie d’assurance « MegaAssur » veut calculer le bonus/malus des conducteurs, ce calcul va tenir compte du nombre d'accidents survenus dans l'année précédente (*nbacc*), ainsi que du  bonus/malus de l’année précédente (*cbmprec*).

Pour cela « MegaAssur » calcule un coefficient d'accident (*coef*) de la manière suivante :

* si le conducteur a eu un ou des accidents *coeff = 1 + 0,05 × nbacc*
* si le conducteur n’a pas eu d’accident, il sera de *0,95*

Le bonus/malus de l’année (*cbm*) correspond à celui de l’année précédente (*cbmprec*) multiplié par le coefficient (*coef*). Mais la valeur du bonus/malus est limitée *3,5* (valeur maximale).  Donc, si le bonus/malus calculé pour un conducteur dépasse cette valeur maximale, il sera égal à  3,5.

Ecrire un programme qui calcule et affiche le bonus/malus d'un conducteur. Ce programme doit demander à l’utilisateur le bonus/malus correspondant à l’année précédente (*cbmprec*), ainsi que le nombre d’accidents (*nbacc*), et après la lecture de ces données  il doit calculer *coef* puis calculer le bonus/malus pour l’année courante et l'afficher.

# Feuille réponse pour la partie I (QCM)

|  |  |
| --- | --- |
| ***Question*** | ***Réponse*** |
| **1** |  |
| **2** |  |
| **3** |  |
| **4** |  |
| **5** |  |
| **6** |  |
| **7** |  |
| **8** |  |
| **9** |  |
| **10** |  |