

Examen d'informatique 1A– Langage C

Vendredi 22 décembre 2017

Durée : 1h30

Aucun document autorisé. Calculatrice interdite.

L'examen est constitué d'une partie QCM (Questions à Choix Multiples) et d'un exercice à rendre sur feuille à part. Pour répondre au QCM, vous devez remplir la feuille de réponse en p.9 et rendre l'ensemble de l'énoncé. Il y a une ou plusieurs réponses possibles à chaque question. Il y a 0,25 point par bonne réponse, 0 point s'il n'y a pas de réponse et -0,25 point par réponse fausse. On supposera que les librairies nécessaires à l'exécution des codes sont incluses dans les morceaux de code fournis.

1. Le(s)quel(s) de ces calculs donne(nt) le bon résultat de la division ?

- a) `int i = 4; int k = 5; int z = i/k;`
- b) `double i = 4; double k = 5; int z = i/k;`
- c) `double i = 4; int k = 5; double z = i/k;`
- d) `int i = 4; int k = 5; int z = (double) i/k;`

2. On déclare les variables suivantes :

```
int a; int b; int c;  
int *p_a; int *p_b; int *p_c;
```

Dans quelle(s) situation(s) n'aura-t-on pas de problème à l'exécution ?

- a) `printf("%x %x %x", &a, &b, &c);`
- b) `scanf("%d", p_a); scanf("%d", p_b);
scanf("%d", p_c);`
- c) `scanf("%d %d %d", p_a, p_b, p_c);`
- d) `scanf("%d %d %d", &a, &b, &c);`

3. Qu'affiche le programme suivant ?

```
#define N 10
int i;
for(i = 0; i < N; i++); {
    if(i % 3 == 0)
        printf("%d, ", i);
}
```

- a) RIEN
- b) 0, 3, 6, 9,
- c) 0, 0, 0, 3, 3, 3, 6, 6, 6, 9,
- d) 0, 0.33, 0.66, 1, 1.33, 1.66, 2, 2.33, 2.66, 3,

4. Soit les initialisations de variables suivantes :

```
int i, int H = 6; int L = 7; int somme = 0;
int tab[6*7];
```

Quel code calcule correctement la somme des composantes du tableau et minimise l'appel à la fonction printf ?

- a)

```
for(i = 0; i < H*L; i++){
    somme = somme + tab[i];
    printf("%d \n", somme);
}
```
- b)

```
int k = H * L;
for(i = 0; i < k; i++){
    somme = somme + tab[i];
    printf("%d \n", somme);
}
```
- c)

```
int k = H * L;
for(i = 0; i < k; i++);
    somme = somme + tab[i];
printf("%d \n", somme);
```
- d)

```
for(i = 0; i < H*L; i++)
    somme = somme + tab[i];
printf("%d \n", somme);
```

5. Soit le code suivant :

```
int Nmin, Ns = 127; Nmin = Ns/60;
```

On veut afficher le nombre de secondes Ns=127 sous le format (2min, 7s) dans la console.

Choisir la/les syntaxe(s) correcte(s) :

- a) `printf("%dmin, %ds\n", Nmin, Ns-Nmin*60);`
- b) `printf("%dmin, %ds\n", Ns/60, Ns%60);`
- c) `printf("%dmin, %s\n", Nmin, Ns-Nmin*60);`

d) `printf("%lfmin, %ds\n",Ns/60.0,Ns%60);`

6. Soit le code suivant :

```
int i,j;
for (i=0;i<10;i++) {
    for (j=0;j<10;j++) {
        if (i<j) printf("1\t");
        else printf("0\t");
    }
    printf("\n");
}
```

Ce programme affiche une matrice :

- a) remplie de 1
- b) contenant des 1 au dessus de la diagonale et des 0 ailleurs
- c) contenant des 1 au dessous de la diagonale et des 0 ailleurs
- d) le programme plante

7. Quel code permet d'obtenir $j=-1$?

```
int i = 0;
int j = 3;
```

a) do

```
{
    j--;
}
while(j>=i) ;
```

c) do

```
{
    j--;
}
while(j>i) ;
```

b) do

```
{
    j--;
}
while(j<i) ;
```

d) while(j>=i)

```
{
    j--;
}
```

8. Soit le tableau suivant :

```
int tab[3] = {1,2,3};
```

Comment afficher la dernière valeur contenue dans tab ?

- a) `printf("%d \n", tab[2]);`
- b) `printf("%d \n", tab[3]);`
- c) `printf("%d \n", *(tab+2));`
- d) `printf("%d \n", *(tab+3));`

9. Soit le tableau suivant :

```
int tab[3] ;
```

Comment remplir la dernière case du tableau ?

- a) `scanf("%d", &tab[2]);`
- b) `scanf("%d", tab+2);`
- c) `scanf("%d", tab+3);`
- d) `scanf("%d", &tab[3]);`

10. Soit la déclaration suivante : `double* p ;` Quelles sont les propositions correctes ?

- a) `*(p + 1)` est un pointeur de type double
- b) `p + 2` est situé 8 octets après `p`
- c) `*p + 1` est de type double
- d) `p + 2` est une adresse

11. Qu'affiche le code suivant ?

```
char chaine[20] = "manger pour vivre";  
printf( ">%s<\n", chaine + 7);  
chaine[6] = '\0';  
printf("%s\n", chaine);  
printf("%c", chaine[12]);
```

- a) `> pour vivre<`
mange
- b) `>pour vivre<`
manger
v
- c) Ce code plante à l'exécution.
- d) `>pour vivre<`
manger\0pour vivre
v

12. Soit la chaîne de caractères

```
char chaine[] = "j'aime l'examen de C";
```

Quelles sont les affirmations correctes :

- a) chaine possède 20 éléments.
- b) L'instruction suivante affiche C : `printf("%c", chaine[20]) ;`
- c) Les 2 instructions suivantes affichent "j'aime l'examen de C++" :
`strcpy(chaine, "++"); printf("%s", chaine) ;`
- d) Les 2 instructions suivantes affichent ++ :

```
strcpy(chaine, "++"); printf("%s",chaine) ;
```

13. Soit N une constante symbolique définie par #define N 10

On déclare un tableau double k[N] que l'on initialise dans le programme. Pour afficher toutes les cases du tableau k, quelle(s) est(sont) la(les) bonne(s) syntaxe(s) ?

- a) printf("%lf ", k);
- b) for(i = 0; i < N-1; i++) printf("%lf ", k[i]);
- c) for(i = 0; i < =N; i++) printf("%lf ", k[i]);
- d) for(i = 0; i < N; i++) printf("%lf ", k[i]);

14. Quelle(s) est(sont) la(les) bonne(s) façon(s) d'initialiser une chaîne de caractères ?

- a) char *c;
strcpy(c, "Bonjour");
- b) char ch[50] = "Bonjour";
- c) char *ch =malloc(50) ; ch= "Bonjour";
- d) char ch[50];
strcpy(ch, "Bonjour");

15. Quel code permet d'allouer un tableau dynamique de 7 réels de type double ?

- a) double tab = malloc(sizeof(double)*7);
- b) double* tab = malloc(28);
- c) double* tab = malloc(56);
- d) aucune réponse n'est correcte

16. Quelle(s) est(sont) le(les) prototype(s) de fonctions qui permettent de retourner les caractères minimum et maximum d'une chaîne de caractères ?

- a) char, char min_max_ch(char *c, int dim);
- b) void min_max_ch(char *c, char min, char max);
- c) void min_max_ch(char *c, char *min, char *max) ;
- d) void min_max_ch(char *c, int dim, int *min, int *max);

17. Voici 4 propositions pour initialiser un tableau de double à 0. Les(la)quelle(s) sont(est) correcte(s) :

- a) void initTab(double *c, int dim){
int i = 0;
c = malloc(dim * sizeof(double));
for(i = 0; i < dim; i++)
c[i] = 0;
}
int main(){
double *z;

```

    initTab(z, 40);
    return 0;
}
b) double* initTab(int dim){
    double* c;
    c = malloc(dim * sizeof(double));
    int i = 0;
    for(i = 0; i < dim; i++)
        c[i] = 0;
    return c;
}
int main(){
    double *z = initTab(40);
    return 0;
}
c) void initTab(double tab[], int dim){
    int i = 0;
    for(i = 0; i < dim; i++)
        c[i] = 0;
}
int main(){
    double tab[40] ;
    initTab(tab,40) ;
    return 0;
}
d) void initTab(double tab[], int dim){
    int i = 0;
    for(i = 0; i < dim; i++)
        c[i] = 0;
}
int main(){
    double* tab=malloc(40*sizeof(double)) ;
    initTab(tab,40) ;
    return 0;
}

```

18. Soit le pointeur de fichier défini par FILE *fp ; on veut ouvrir le fichier C:/Mesdocuments/essai.txt en mode lecture. Quelles sont les propositions correctes ?

- a) fp=fopen("C:/Mesdocuments/essai.txt", "r") ;
- b) fp=fopen(C:/Mesdocuments/essai.txt, "r") ;
- c) le fichier C:/Mesdocuments/essai.txt doit exister.

d) Si le fichier C:/Mesdocuments/essai.txt n'existe pas, il est créé.

19. Soit un fichier au format texte repéré par son pointeur de fichier fp contenant les lignes suivantes :

```
bonjour
x=8
y=10
```

On souhaite lire ce fichier et stocker les valeurs numériques (c'est-à-dire 8 et 10) dans deux variables du programme.

On suppose que l'ouverture s'est correctement déroulée. On déclare la chaîne suivante :
`char chaine[10] ;`

Quelles sont les propositions correctes ?

- a) `int tab[2] ; fscanf(fp,"%s",chaine) ;
fscanf(fp,"x=%d\n",tab) ;
fscanf(fp,"y=%d\n",tab+1) ;`
- b) `int tab[2] ; fscanf(fp,"%s",chaine) ;
fscanf(fp,"x=%d\n",*tab) ;
fscanf(fp,"y=%d\n",*tab+1) ;`
- c) `int tab[2] ;
fscanf(fp,"%c",chaine) ;
fscanf(fp,"x=%d\n",tab) ;
fscanf(fp,"y=%d\n",tab+1) ;`
- d) `int a,b ;
fscanf(fp,"%s",chaine) ;
fscanf(fp,"x=%d\n",&a) ;
fscanf(fp,"y=%d\n",&b) ;`

20. Quelles sont les affirmations correctes :

- a) La commande build permet de générer un fichier exécutable
- b) La compilation génère un fichier exécutable
- c) Un fichier objet contient du langage machine
- d) Un fichier d'entête .h doit être compilé

Exercice à rendre sur une feuille à part :

En C on peut définir un nombre complexe avec deux réels de type double, l'un pour la partie réelle et le second pour la partie imaginaire. Le nombre imaginaire i sera par exemple représenté par deux réels, l'un valant 0 et le second valant 1.

- 1) Faire une fonction qui permet à l'utilisateur de renseigner la partie réelle puis la partie imaginaire d'un nombre complexe. **1 point**
- 2) Faire une fonction qui prend comme arguments d'entrée la partie réelle et la partie imaginaire d'un nombre complexe et qui affiche à l'écran le nombre complexe sous la forme « $x + i*y$ ». **1 point**
- 3) Faire une fonction qui retourne le module au carré d'un nombre complexe. **1 point**
- 4) Faire une fonction qui échange la partie réelle et la partie imaginaire d'un nombre complexe. Ecrire les lignes d'instruction permettant de vérifier le bon fonctionnement de cette fonction. **2 points**
- 5) Déclarer un tableau dynamique de n parties réelles et un tableau dynamique de n parties imaginaires. **1 point**
- 6) Faire une fonction qui calcule le module au carré de n nombres complexes. **1 point**
- 7) On souhaite utiliser la modularité inter-fichiers pour traiter les questions 1) à 6). Décrire la structure du projet en donnant le nom des fichiers à créer (pensez à bien spécifier leurs extensions) et écrire ce qu'il faut mettre dans chacun de ces fichiers. **3 points**

NOM :

PRENOM :

Feuille de réponses au QCM à rendre avec votre copie.

Cochez les cases à droite du numéro de vos réponses.

Q1		Q6		Q11		Q16	
1		1		1		1	
2		2	x	2	x	2	
3	x	3		3		3	x
4		4		4		4	
Q2		Q7		Q12		Q17	
1	x	1	x	1		1	
2		2		2		2	X
3		3		3		3	X
4	x	4	x	4	x	4	X
Q3		Q8		Q13		Q18	
1	x	1	x	1		1	X
2		2		2		2	
3		3	x	3		3	x
4		4		4	x	4	
Q4		Q9		Q14		Q19	
1		1	X	1		1	x
2		2	x	2	x	2	
3		3		3	x	3	
4	x	4		4	x	4	x
Q5		Q10		Q15		Q20	
1	x	1		1		1	x
2	x	2		2		2	
3		3	X	3	x	3	x
4		4	x	4		4	