



NOM :

PRÉNOM :

DATE DE NAISSANCE :

**Questions de cours (2 pts) :**

1) Quelles variables sont créées au moment de la compilation, sont initialisées par défaut à zéro et existent pendant toute la durée du programme ? (cochez la ou les bonnes réponses)

Les variables globales     Les variables locales     Les variables automatiques     Les variables statiques

2) Si `t` est un tableau à 2 dimensions, quelle expression est équivalente à `t[i][j]` (cochez la bonne réponse) ?

`** (t+i+j)`     `* (t[i]+j)`     `* (*t+i)+j`     `* (*t[i]+j)`     `**t+i+j`     Aucune de ces 5 réponses.

3) Quelle fonction est déconseillée et a été supprimée des nouvelles normes du langage C (cochez la bonne réponse) ?

`getchar`     `fgets`     `gets`     `strcpy`     `printf`     une autre (précisez laquelle) : .....

4) Quelle est la valeur de l'expression `sizeof(char)` (cochez la bonne réponse) ?

0     1     2     4     cela dépend de la machine et du compilateur

**Exercice 1 – Programme (8 pts) :**

Voir le code source du programme sur la page Annexe.

1) Le programme donné est garanti sans erreurs. Qu'affiche-t-il ?

2) La ligne 17 affecte zéro à la case numéro `n` du tableau. Pourquoi ? Cochez la bonne réponse (une seule) :

- Pour que le dernier nombre soit multiplié par 10.
- Pour que la chaîne de caractère se termine correctement.
- Pour que le dernier caractère affiché soit un zéro.
- Pour rien, on peut enlever cette ligne, elle est inutile.
- Aucune des réponses ci-dessus n'est correcte.

3) À la ligne 11, il y a une variable `s`. Qu'est-ce que c'est ? Cochez la bonne réponse (une seule) :

- `s` est une *variable globale*, son type est *int*.
- `s` est un *paramètre effectif*, son type est *tableau de int*.
- `s` est un *paramètre formel*, son type est *pointeur sur int*.
- `s` est une *constante*, son type est *tableau de int*.
- Aucune des réponses ci-dessus n'est correcte.

**Exercice 2 – Jeu du Taquin (10 pts) :**

Voir les explications sur le programme du Taquin dans la page Annexe.

1) Comment initialiser correctement le tableau ? Cochez la bonne réponse (une seule) :

- `int taquin[3][3]={{1, 2, 3, 4}, {5, 6, 7, 8}, {9, 10, 11, 12}, {13, 14, 15}};`
- `int taquin[][3]={{1, 2, 3, 4}, {5, 6, 7, 8}, {9, 10, 11, 12}, {13, 14, 15, 0}};`
- `int taquin[][4]={{1, 2, 3, 4}, {5, 6, 7, 8}, {9, 10, 11, 12}, {13, 14, 15}};`
- `int taquin[4][4]={{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 0}};`
- Aucune des réponses ci-dessus n'est correcte.



NOM :

PRÉNOM :

DATE DE NAISSANCE :

2) Pour chercher et renvoyer la position de la place vide, la fonction suivante est définie :

```
void position_vide(int taq[4][4], int *i, int *j){  
    for(*i=0;*i<4;++*i)  
        for(*j=0;*j<4;++*j)  
            if(.....)  
                return;  
}
```

Complétez la condition du test en cochant la bonne réponse (une seule) :

- |   |  |
|---|--|
| <input type="checkbox"/> if(taq[i][j]==0)   | <input type="checkbox"/> if(*(taq+i)+j)==0)                            |
| <input type="checkbox"/> if(taq[*i][*j]!=0) | <input type="checkbox"/> if(taq[*i][*j]==0)                            |
|   | <input type="checkbox"/> Aucune des réponses ci-dessus n'est correcte. |

3) La fonction ci-dessous permet de déplacer horizontalement le carreau qui est juste à gauche (sens = +1) ou à droite (sens = -1) de la place vide (sa valeur de retour permet de savoir si elle a réussi) :

```
int glisser_horiz(int taq[4][4], int sens){ //sens : -1=gauche, +1=droite  
    int i,j;  
    position_vide(taq,&i,&j);  
    if((j==0 && sens==1) || (j==3 && sens==-1))  
        return 0; // déplacement impossible  
    .....  
    taq[i][j-sens]=0;  
    return 1; // déplacement réussi  
}
```

Quelle est l'affectation qu'il faut faire à la 6<sup>e</sup> ligne ? Cochez la bonne réponse (une seule) :

- taq[i][j+sens]=taq[i][j];
- taq[i][j-sens]=taq[i][j+sens];
- taq[i][j]=taq[i-sens][j];
- taq[i][j]=taq[i][j-sens];
- taq[i][j+sens]=taq[i][j-sens];
- Aucune des réponses ci-dessus n'est correcte.

4) Voici la fonction que l'on a écrit pour savoir si le joueur a gagné :

```
int c_est_gagne(int taq[4][4]){  
    for(int *p=&taq[0][0], n=1; n<16; ++n)  
        if( ..... )  
            return 0;  
    return 1;  
}
```

Il manque la condition du test. Que peut-on mettre ? Cochez la bonne réponse (une seule) :

- |  |  |
|--|--|
| <input type="checkbox"/> if(*p++!=n)   | <input type="checkbox"/> if(++p!=n)                                    |
| <input type="checkbox"/> if((*p)++!=n) | <input type="checkbox"/> if(p++!=&n)                                   |
| <input type="checkbox"/> if(++*p!=n)   | <input type="checkbox"/> Aucune des réponses ci-dessus n'est correcte. |



Page Annexe des exercices

Pour chaque exercice, lisez d'abord cette page puis répondez aux questions des pages 1 et 2

Exercice 1 – Programme :

Observez attentivement le code source de ce programme en C.

```
1 #include <stdio.h>
2 #include <ctype.h>
3 void linearise(int t[][2], int s[]){
4     for(int i=0, n=0; i<=0; n++) {
5         s[n] = t[i][0];
6         printf("%d ",s[n]);
7         i = (t[i][1]==0)?(-1):(t[i][1]);
8     }
9     printf("\n");
10 }
11 void decode(int n, int s[n],char ch[n+1]){
12     for(int i=0; i<n; ++i){
13         ch[i] = (s[i]>0)?(s[i]-1+'A'):(-'0'-s[i]);
14         if(!isalpha(ch[i]) && !isdigit(ch[i]))
15             ch[i] = '_'; // tiret de soulignement
16     }
17     ch[n] = 0;
18 }
19 int main(void){
20     int liste[6][2] = {{3,1},{-10,4},{14},{-1,5},{2,3},{5,2}};
21     int seq[6];
22     linearise(liste,seq);
23     int secret[6]={-10,2,18,1,22}; // que 5 nombres !?
24     char message[7];
25     decode(6,secret,message);
26     puts(message);
27     char chaine[7];
28     decode(6,seq,chaine);
29     puts(chaine);
30 }
```

Rappels :

*isalpha(c)* permet de savoir si le caractère *c* est une lettre et *isdigit(c)* si c'est un chiffre.

L'instruction *var = condition ? v1 : v2;* est équivalente à

*if(condition) var=v1; else var=v2;*

Exercice 2 – Jeu du Taquin :

Le **Taquin** (en anglais *Fifteen-Puzzle*) est un jeu créé au 19<sup>e</sup> siècle composé de quinze petits carreaux numérotés de 1 à 15 qui glissent dans un cadre en bois (ou en plastique) carré. Le cadre peut en prendre 16 mais une place est laissée vide pour pouvoir déplacer les carreaux en les glissant horizontalement ou verticalement. La figure ci-contre montre les carreaux dans l'ordre. Pour jouer, on commence par mélanger les carreaux (en les glissant aléatoirement), puis le but du jeu est de les remettre dans l'ordre.

On veut écrire en C un programme qui permet de jouer au Taquin sur l'ordinateur. Pour cela on utilise un tableau d'entiers de 4 lignes et 4 colonnes que l'on initialise avec les nombres de 1 à 15. On met zéro dans la dernière case correspondant à la place vide.





NOM :

PRÉNOM :

DATE DE NAISSANCE :

### Questions de cours (2 pts) :

1) Quelles variables sont créées au moment de la compilation, sont initialisées par défaut à zéro et existent pendant toute la durée du programme ? (cochez la ou les bonnes réponses)

Les variables globales     Les variables locales     Les variables automatiques     Les variables statiques

2) Si t est un tableau à 2 dimensions, quelle expression est équivalente à  $t[i][j]$  (cochez la bonne réponse) ?

$**(t+i+j)$       $*(t[i]+j)$       $*(t+i)+j$       $*(t[i]+j)$       $**t+i+j$      Aucune de ces 5 réponses.

3) Quelle fonction est déconseillée et a été supprimée des nouvelles normes du langage C (cochez la bonne réponse) ?

getchar     fgets     gets     strcpy     printf     une autre (précisez laquelle) : .....

4) Quelle est la valeur de l'expression `sizeof(char)` (cochez la bonne réponse) ?

0     1     2     4     cela dépend de la machine et du compilateur

### Exercice 1 – Programme (8 pts) :

Voir le code source du programme sur la page Annexe.

1) Le programme donné est garanti sans erreurs. Qu'affiche-t-il ?

```
3  -10  2  -1  5  14
_BRAV0
C_B1EN
```

2) La ligne 17 affecte zéro à la case numéro n du tableau. Pourquoi ? Cochez la bonne réponse (une seule) :

- Pour que le dernier nombre soit multiplié par 10.
- Pour que la chaîne de caractère se termine correctement.
- Pour que le dernier caractère affiché soit un zéro.
- Pour rien, on peut enlever cette ligne, elle est inutile.
- Aucune des réponses ci-dessus n'est correcte.

3) À la ligne 11, il y a une variable s. Qu'est-ce que c'est ? Cochez la bonne réponse (une seule) :

- s est une *variable globale*, son type est *int*.
- s est un *paramètre effectif*, son type est *tableau de int*.
- s est un *paramètre formel*, son type est *pointeur sur int*.
- s est une *constante*, son type est *tableau de int*.
- Aucune des réponses ci-dessus n'est correcte.

### Exercice 2 – Jeu du Taquin (10 pts) :

Voir les explications sur le programme du Taquin dans la page Annexe.

1) Comment initialiser correctement le tableau ? Cochez la bonne réponse (une seule) :

- `int taquin[3][3]={{1, 2, 3, 4}, {5, 6, 7, 8}, {9, 10, 11, 12}, {13, 14, 15}};`
- `int taquin[][3]={{1, 2, 3, 4}, {5, 6, 7, 8}, {9, 10, 11, 12}, {13, 14, 15, 0}};`
- `int taquin[][4]={{1, 2, 3, 4}, {5, 6, 7, 8}, {9, 10, 11, 12}, {13, 14, 15}};`
- `int taquin[4][4]={{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 0}};`
- Aucune des réponses ci-dessus n'est correcte.



NOM :

PRÉNOM :

DATE DE NAISSANCE :

2) Pour chercher et renvoyer la position de la place vide, la fonction suivante est définie :

```
void position_vide(int taq[4][4], int *i, int *j){  
    for(*i=0;*i<4;++*i)  
        for(*j=0;*j<4;++*j)  
            if(.....)  
                return;  
}
```

Complétez la condition du test en cochant la bonne réponse (une seule) :

- |   |  |
|---|--|
| <input type="checkbox"/> if(taq[i][j]==0)   | <input type="checkbox"/> if(*(taq+i)+j)==0)                            |
| <input type="checkbox"/> if(taq[*i][*j]!=0) | <input checked="" type="checkbox"/> if(taq[*i][*j]==0)                 |
|   | <input type="checkbox"/> Aucune des réponses ci-dessus n'est correcte. |

3) La fonction ci-dessous permet de déplacer horizontalement le carreau qui est juste à gauche (sens = +1) ou à droite (sens = -1) de la place vide (sa valeur de retour permet de savoir si elle a réussi) :

```
int glisser_horiz(int taq[4][4], int sens){ //sens : -1=gauche, +1=droite  
    int i,j;  
    position_vide(taq,&i,&j);  
    if((j==0 && sens==1) || (j==3 && sens==-1))  
        return 0; // déplacement impossible  
    .....  
    taq[i][j-sens]=0;  
    return 1; // déplacement réussi  
}
```

Quelle est l'affectation qu'il faut faire à la 6<sup>e</sup> ligne ? Cochez la bonne réponse (une seule) :

- taq[i][j+sens]=taq[i][j];
- taq[i][j-sens]=taq[i][j+sens];
- taq[i][j]=taq[i-sens][j];
- taq[i][j]=taq[i][j-sens];
- taq[i][j+sens]=taq[i][j-sens];
- Aucune des réponses ci-dessus n'est correcte.

4) Voici la fonction que l'on a écrit pour savoir si le joueur a gagné :

```
int c_est_gagne(int taq[4][4]){  
    for(int *p=&taq[0][0], n=1; n<16; ++n)  
        if( ..... )  
            return 0;  
    return 1;  
}
```

Il manque la condition du test. Que peut-on mettre ? Cochez la bonne réponse (une seule) :

- |   |  |
|---|--|
| <input checked="" type="checkbox"/> if(*p++!=n) | <input type="checkbox"/> if(++p!=n)                                    |
| <input type="checkbox"/> if((*p)++!=n)          | <input type="checkbox"/> if(p++!=&n)                                   |
| <input type="checkbox"/> if(++*p!=n)            | <input type="checkbox"/> Aucune des réponses ci-dessus n'est correcte. |



Page Annexe des exercices

Pour chaque exercice, lisez d'abord cette page puis répondez aux questions des pages 1 et 2

Exercice 1 – Programme :

Observez attentivement le code source de ce programme en C.

```
1 #include <stdio.h>
2 #include <ctype.h>
3 void linearise(int t[][2], int s[]){
4     for(int i=0, n=0; i<=0; n++) {
5         s[n] = t[i][0];
6         printf("%d ",s[n]);
7         i = (t[i][1]==0)?(-1):(t[i][1]);
8     }
9     printf("\n");
10 }
11 void decode(int n, int s[n],char ch[n+1]){
12     for(int i=0; i<n; ++i){
13         ch[i] = (s[i]>0)?(s[i]-1+'A'):(-'0'-s[i]);
14         if(!isalpha(ch[i]) && !isdigit(ch[i]))
15             ch[i] = '_'; // tiret de soulignement
16     }
17     ch[n] = 0;
18 }
19 int main(void){
20     int liste[6][2] = {{3,1},{-10,4},{14},{-1,5},{2,3},{5,2}};
21     int seq[6];
22     linearise(liste,seq);
23     int secret[6]={-10,2,18,1,22}; // que 5 nombres !?
24     char message[7];
25     decode(6,secret,message);
26     puts(message);
27     char chaine[7];
28     decode(6,seq,chaine);
29     puts(chaine);
30 }
```

Rappels :

*isalpha(c)* permet de savoir si le caractère *c* est une lettre et *isdigit(c)* si c'est un chiffre.

L'instruction *var = condition ? v1 : v2;* est équivalente à

*if(condition) var=v1; else var=v2;*

Exercice 2 – Jeu du Taquin :

Le **Taquin** (en anglais *Fifteen-Puzzle*) est un jeu créé au 19<sup>e</sup> siècle composé de quinze petits carreaux numérotés de 1 à 15 qui glissent dans un cadre en bois (ou en plastique) carré. Le cadre peut en prendre 16 mais une place est laissée vide pour pouvoir déplacer les carreaux en les glissant horizontalement ou verticalement. La figure ci-contre montre les carreaux dans l'ordre. Pour jouer, on commence par mélanger les carreaux (en les glissant aléatoirement), puis le but du jeu est de les remettre dans l'ordre.

On veut écrire en C un programme qui permet de jouer au Taquin sur l'ordinateur. Pour cela on utilise un tableau d'entiers de 4 lignes et 4 colonnes que l'on initialise avec les nombres de 1 à 15. On met zéro dans la dernière case correspondant à la place vide.

