Université de Tlemcen
Faculté des Sciences
1re année Licence MI

Examen de Rattrapage d'Algorithmique2

Année 2022-2023
Semestre 2
18/06/2023
Durée : 1h30

NOM : PRÉNOM :

DATE DE NAISSANCE :

Questions de cours (2,5 points)

1) Si la variable s correspond à une chaîne de caractères correctement formée,

a) Qu'affiche l'instruction suivante ? `printf("%d",s[strlen(s)]);`

☐ le code ASCII du 1er caractère de la chaîne ☐ le code ASCII du dernier caractère
☐ le code ASCII du caractère `'\n'` ☐ 0 (zéro) ☐ rien, une erreur ou une valeur indéfinie

b) Si `strlen(s)` donne 30, comment sera la valeur donnée par `sizeof(s)` ? (une seule réponse)

☐ = 30 ☐ < 30 ☐ > 30
☐ < 30 si s est un pointeur, sinon > 30 ☐ > 30 si s est un pointeur, sinon < 30

c) Que fait l'instruction `*s=0;` ? (une seule réponse)

☐ change le 1er caractère de la chaîne en caractère '0' ☐ change le 1er caractère de la chaîne en caractère 'A'
☐ efface le 1er caractère de la chaîne et strlen(s) donnera 29 au lieu de 30
☐ si s est un tableau, la chaîne de caractère sera vide et strlen(s) donnera 0

2) Soit les structures suivantes :

```
struct enseignant{
      char nom[30], prenom[30];
      int annee_recrutement;
} x = {"Sharif","Ali",2008};
```

```
struct matiere{
      char nom[30], filiere[30];
      int semestre, coefficient;
      struct enseignant * responsable;
} a = {"Programmation","Licence Inform.",1,5};
```

a) Comment affecter l'enseignant x à la matière a ? (une seule réponse)

☐ `responsable(matiere a) = &x;` ☐ `a.responsable = x;`
☐ `a.responsable = &x;` ☐ `x = a->responsable;`
☐ `(*a).responsable = x;` ☐ `a->responsable = &x;`

b) Comment afficher le prénom de l'enseignant chargé de la matière a ? (une seule réponse)

☐ `puts(a.enseignant.prenom);` ☐ `puts(a->responsable.prenom);`
☐ `puts(a.responsable->prenom);` ☐ `puts((*a).responsable.prenom);`
☐ `puts((*a.enseignant).prenom);` ☐ `puts(a.responsable.prenom);`

c) Comment changer le prénom de l'enseignant chargé de la matière a en « Omar » ? (une seule réponse)

☐ `a->responsable.prenom[]="Omar";` ☐ `strstr((*a.responsable).prenom,"Omar");`
☐ `a.enseignant.prenom="Omar";` ☐ `strcpy((*a).responsable.prenom,"Omar");`
☐ `sscanf(a->enseignant->prenom,"Omar");` ☐ `sprintf(a.responsable->prenom,"Omar");`

Exercice 1 : Tic-tac-toe (7,5 points)

Le programme source en page annexe 1 est un jeu appelé « tic-tac-toe », où il manque 10 petits morceaux de code.

1) Indiquez sous chacun le numéro de ligne où il faut le placer pour compléter le programme :

Morceau de code	`int`	`i%2+1`	`t[--i][--j]`	`!vainqueur`	`r[t[i][j]]`
Numéro de ligne					
Morceau de code	`joueur`	`r[k]`	`r[joueur]`	`vainqueur`	`k!=0`
Numéro de ligne					

2) À la ligne 2, la variable r est... (une seule bonne réponse) :

☐ une variable statique ☐ une chaîne de caractères correctement formée
☐ un tableau de pointeurs ☐ une variable locale ☐ un pointeur ☐ aucune des 5 réponses

3) À la ligne 3 (<u>une seule</u> bonne réponse) :

- ☐ t est un paramètre formel de type pointeur qui contiendra l'adresse de la 1re ligne du tableau t de main
- ☐ t est un paramètre effectif de type tableau qui contiendra les valeurs du tableau t de main
- ☐ les dimensions du paramètre effectif t peuvent être enlevées : on peut écrire int t[][]
- ☐ la 2e dimension du paramètre effectif t peut être enlevée : on peut écrire int t[3][] ☐ aucune des 4 réponses

4) La ligne 35 peut être remplacée par les lignes suivantes sauf <u>une</u>, laquelle ? (cochez une seule réponse) :

- ☐ `int t[3][3]={{0,0,0}};`
- ☐ `int t[3][3]={{0},{0},{0}};`
- ☐ `int t[][3]={{0,0,0},{0,0,0},{0,0,0}};`
- ☐ `int t[][3]={{0},{0},{0}};`
- ☐ `int t[3][3];`
- ☐ Aucune (elles sont toutes correctes)

Exercice 2 : Décodage <u>(10 points)</u>

Examinez attentivement le programme en C de la page annexe 2 avant de répondre aux questions.

1) La fonction init_s de ce programme est équivalente aux fonctions suivantes (on peut la remplacer par ces fonctions sans que cela ne change le résultat), sauf une, laquelle ?

☐	`void init_s1(int *s, int m){`
	`  if(m>0)`
	`  do`
	`  s[--m]=1;`
	`  while(m>0);`
	`}`
		☐	`void init_s3(int s[], int m){`
			`  for(int *p=s;p<s+m;++p)`
			`  *p=1;`
			`}`

☐	`void init_s(int s[], int m){`
	`  for(int i=m;i>0;--i)`
	`  s[i]++=1;`
	`}`
		☐	`void init_s(int *s, int m){`
			`  for(int i=0;i<m;++i)`
			`  s[i]=1;`
			`}`

☐ Aucune (les fonctions sont toutes équivalentes)			

2) Par quelles instructions exactement équivalentes peut-on remplacer la ligne 32 ? (une seule réponse)

- ☐ `lp[++np]=2;`
- ☐ `np++; lp[np]=2;`
- ☐ `++np; lp[np]=2;`
- ☐ `lp[np]=2;++np;`
- ☐ `++lp[np]=2;`
- ☐ `aucune des 5 réponses`

3) A la ligne 41, que représente lp ? (une réponse)

- ☐ La variable statique locale lp est un pointeur
- ☐ La variable lp n'est pas un pointeur mais l'expression lp de la ligne 41 donne l'adresse de lp[0]
- ☐ L'adresse de lp[1], le 1er élément de lp, est le paramètre formel qui sera transmis à la fonction fill_lp
- ☐ C'est un paramètre formel global, alors qu'à la ligne 30 lp est un paramètre effectif local
- ☐ Aucune réponse n'est correcte

4) Qu'affiche ce programme ?

.....

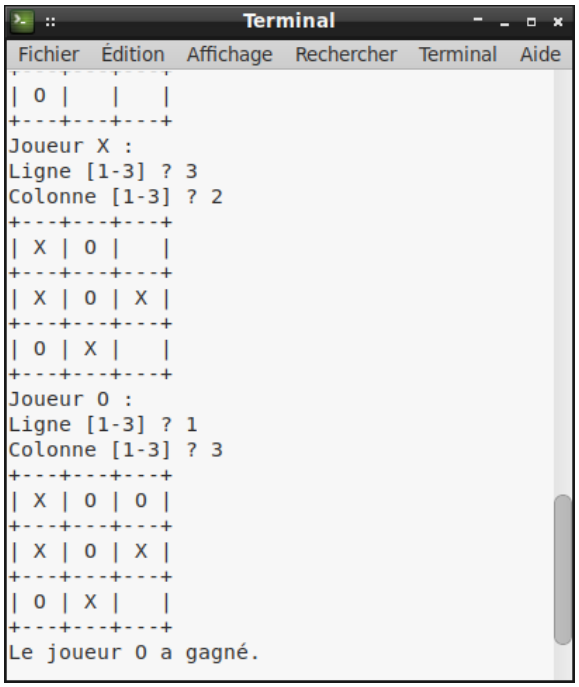
.....

Page annexe 1 : Programme de l'exercice 1 (Jeu Tic-tac-toe)

Le « tic-tac-toe », aussi appelé « morpion » ou « X et O » est un jeu simple comportant une grille de 3×3 cases. Chacun des deux joueurs, appelés « X » et « O » joue son tour en inscrivant son symbole dans une case libre. Le premier qui inscrit son symbole sur trois cases alignées, verticalement, horizontalement ou en diagonale gagne la partie. Si les neuf cases sont remplies sans qu'aucun alignement ne soit obtenu, la partie est nulle.

Le joueur O a gagné

```c
#include <stdio.h>
char r[3]={' ','X','O'};
void affiche(int t[3][3]){
    puts("+---+---+---+");
    for(int i=0 ; i<3 ; ++i){
        for (int j=0 ; j<3 ; ++j)
            printf("| %c ", ..........);
        puts("|\n+---+---+---+");
    }
}
........ joue(int t[3][3], int joueur){
    printf("Joueur %c :\n",..........);
    int i, j, k;
    do {
        do {
            printf("Ligne [1-3] ? ");
            scanf("%d",&i);
        } while(i<1 || i>3);
        do {
            printf("Colonne [1-3] ? ");
            scanf("%d",&j);
        } while(j<1 || j>3);
        k = ..........;
        if(k!=0)
            printf("Cette case contient un %c ! Recommencez.\n", .........);
    } while(.........);
    t[i][j] = .........;
    return ((t[i][0]==joueur && t[i][1]==joueur && t[i][2]==joueur)
          ||(t[0][j]==joueur && t[1][j]==joueur && t[2][j]==joueur)
          ||(t[0][0]==joueur && t[1][1]==joueur && t[2][2]==joueur)
          ||(t[2][0]==joueur && t[1][1]==joueur && t[0][2]==joueur));
}
int main(void){
    puts("-----------\nTic-Tac-Toe\n-----------\n");
    int t[3][3] = {{0}};
    affiche(t);
    int vainqueur = 0;
    for(int i=0 ; i<9 && .........; ++i){
        int joueur = .........;
        vainqueur = joue(t, joueur)*joueur;
        affiche(t);
    }
    if(.........)
        printf("Le joueur %c a gagné.\n",r[vainqueur]);
    else
        puts("Match nul.");
}
```



Exemple d'exécution (fin de partie)

Page annexe 2 : Programme de l'exercice 2 (Décodage)

```
 1 #include<stdio.h>
 2 #include<string.h>
 3 void fill_s(int *s, int m){
 4     s[0]=s[1]=0;
 5     for(int n=2;n*n<m;++n)
 6         if(s[n])
 7             for(int i=n*n;i<m;i+=n)
 8                 s[i]=0;
 9 }
10 void init_s(int s[], int m){
11     while(m--)
12         *s++=1;
13 }
14 void decode1(char *text){
15     int n=strlen(text);
16     int s[n];
17     init_s(s,n);
18     fill_s(s,n);
19     for(int i=0;i<n;++i)
20         if(s[i])
21             putchar(text[i]);
22     puts("");
23 }
24 int ispn(int *sp, int n){
25     for(int j=0;sp[j]*sp[j]<=n;j++)
26         if (n%sp[j]==0)
27             return 0;
28     return 1;
29 }
30 void fill_lp(int *lp, int n){
31     int np=0;
32     lp[np++]=2;
33     for(int i=3;i<n;i+=2)
34         if(ispn(lp,i))
35             lp[np++]=i;
36     lp[np]=0;
37 }
38 void decode2(char *text){
39     int n=strlen(text);
40     int lp[n/2];
41     fill_lp(lp,n);
42     for(int i=0;lp[i]>0 && lp[i]<n;++i)
43         putchar(text[lp[i]]);
44     puts("");
45 }
46 int main(void){
47     char text1[]="Hier attendons cet homme qui n'est pas la";
48     decode1(text1);
49     char text2[]="Mad.brains,tacit.hair.beauty";
50     decode2(text2);
51 }
```

Université de Tlemcen
Faculté des Sciences
1re année Licence MI

Corrigé-type de l'Examen de Rattrapage d'Algorithmique2

Année 2022-2023
Semestre 2
18/06/2023
Durée : 1h30

Questions de cours (2,5 points)

1) Si la variable s correspond à une chaîne de caractères correctement formée,

a) Qu'affiche l'instruction suivante ? `printf("%d",s[strlen(s)]);`

☐ le code ASCII du 1er caractère de la chaîne ☐ le code ASCII du dernier caractère
☐ le code ASCII du caractère `'\n'` ☒ **0 (zéro)** ☐ rien, une erreur ou une valeur indéfinie

b) Si `strlen(s)` donne 30, comment sera la valeur donnée par `sizeof(s)` ? (une seule réponse)

☐ = 30 ☐ < 30 ☐ > 30
☒ **< 30 si s est un pointeur, sinon > 30** ☐ > 30 si s est un pointeur, sinon < 30

c) Que fait l'instruction `*s=0;` ? (une seule réponse)

☐ change le 1er caractère de la chaîne en caractère '0' ☐ change le 1er caractère de la chaîne en caractère 'A'
☐ efface le 1er caractère de la chaîne et strlen(s) donnera 29 au lieu de 30
☒ **si s est un tableau, la chaîne de caractère sera vide et strlen(s) donnera 0**

2) Soit les structures suivantes :

```
struct enseignant{
      char nom[30], prenom[30];
      int annee_recrutement;
} x = {"Sharif","Ali",2008};
```

```
struct matiere{
      char nom[30], filiere[30];
      int semestre, coefficient;
      struct enseignant * responsable;
} a = {"Programmation","Licence Inform.",1,5};
```

a) Comment affecter l'enseignant x à la matière a ? (une seule réponse)

☐ `responsable(matiere a) = &x;` ☐ `a.responsable = x;`
☒ **`a.responsable = &x;`** ☐ `x = a->responsable;`
☐ `(*a).responsable = x;` ☐ `a->responsable = &x;`

b) Comment afficher le prénom de l'enseignant chargé de la matière a ? (une seule réponse)

☐ `puts(a.enseignant.prenom);` ☐ `puts(a->responsable.prenom);`
☒ **`puts(a.responsable->prenom);`** ☐ `puts((*a).responsable.prenom);`
☐ `puts((*a.enseignant).prenom);` ☐ `puts(a.responsable.prenom);`

c) Comment changer le prénom de l'enseignant chargé de la matière a en « Omar » ? (une seule réponse)

☐ `a->responsable.prenom[]="Omar";` ☐ `strstr((*a.responsable).prenom,"Omar");`
☐ `a.enseignant.prenom="Omar";` ☐ `strcpy((*a).responsable.prenom,"Omar");`
☐ `sscanf(a->enseignant->prenom,"Omar");` ☒ **`sprintf(a.responsable->prenom,"Omar");`**

Exercice 1 : Tic-tac-toe (7,5 points)

Le programme source en page annexe 1 est un jeu appelé « tic-tac-toe », où il manque 10 petits morceaux de code.

1) Indiquez sous chacun le numéro de ligne où il faut le placer pour compléter le programme :

Morceau de code	`int`	`i%2+1`	`t[--i][--j]`	`!vainqueur`	`r[t[i][j]]`
Numéro de ligne	**11**	**39**	**23**	**38**	**7**
Morceau de code	`joueur`	`r[k]`	`r[joueur]`	`vainqueur`	`k!=0`
Numéro de ligne	**27**	**25**	**12**	**43**	**26**

2) À la ligne 2, la variable r est... (une seule bonne réponse) :

☒ **une variable statique** ☐ une chaîne de caractères correctement formée
☐ un tableau de pointeurs ☐ une variable locale ☐ un pointeur ☐ aucune des 5 réponses

3) À la ligne 3 (une seule bonne réponse) :

- ☒ **t est un paramètre formel de type pointeur qui contiendra l'adresse de la 1re ligne du tableau t de main**
- ☐ t est un paramètre effectif de type tableau qui contiendra les valeurs du tableau t de main
- ☐ les dimensions du paramètre effectif t peuvent être enlevées : on peut écrire int t[][]
- ☐ la 2e dimension du paramètre effectif t peut être enlevée : on peut écrire int t[3][]
- ☐ aucune des 4 réponses

4) La ligne 35 peut être remplacée par les lignes suivantes sauf une, laquelle ? (cochez une seule réponse) :

- ☐ `int t[3][3]={{0,0,0}};`
- ☐ `int t[3][3]={{0},{0},{0}};`
- ☐ `int t[][3]={{0,0,0},{0,0,0},{0,0,0}};`
- ☐ `int t[][3]={{0},{0},{0}};`
- ☒ `int t[3][3];`
- ☐ Aucune (elles sont toutes correctes)

Exercice 2 : Décodage (10 points)

Examinez attentivement le programme en C de la page annexe 2 avant de répondre aux questions.

1) La fonction init_s de ce programme est équivalente aux fonctions suivantes (on peut la remplacer par ces fonctions sans que cela ne change le résultat), sauf une, laquelle ?

☐	`void init_s(int *s, int m){`
	`  if(m>0)`
	`  do`
	`  s[--m]=1;`
	`  while(m>0);`
	`}`
		☐	`void init_s(int s[], int m){`
			`  for(int *p=s;p<s+m;++p)`
			`  *p=1;`
			`}`

☒	`void init_s(int s[], int m){`
	`  for(int i=m;i>0;--i)`
	`  s[i]++=1;`
	`}`
		☐	`void init_s(int *s, int m){`
			`  for(int i=0;i<m;++i)`
			`  s[i]=1;`
			`}`

☐ Aucune (les fonctions sont toutes équivalentes)			

2) Par quelles instructions exactement équivalentes peut-on remplacer la ligne 32 ? (une seule réponse)

- ☐ `lp[++np]=2;`
- ☐ `np++; lp[np]=2;`
- ☐ `++np; lp[np]=2;`
- ☒ `lp[np]=2;++np;`
- ☐ `++lp[np]=2;`
- ☐ `aucune des 5 réponses`

3) A la ligne 41, que représente lp ? (une réponse)

- ☐ La variable statique locale lp est un pointeur
- ☒ **La variable lp n'est pas un pointeur mais l'expression lp de la ligne 41 donne l'adresse de lp[0]**
- ☐ L'adresse de lp[1], le 1er élément de lp, est le paramètre formel qui sera transmis à la fonction fill_lp
- ☐ C'est un paramètre formel global, alors qu'à la ligne 30 lp est un paramètre effectif local
- ☐ Aucune réponse n'est correcte

4) Qu'affiche ce programme ?

eratosthenes

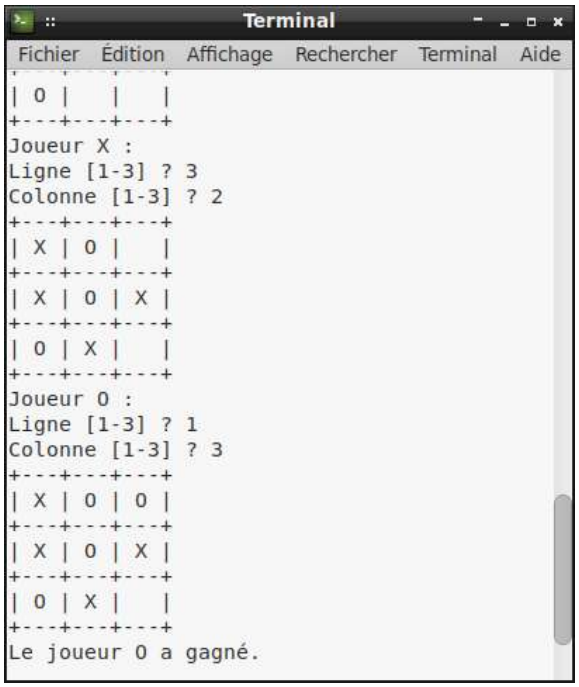
d.ritchie

Page annexe 1 : Programme de l'exercice 1 (Jeu Tic-tac-toe)

Le « tic-tac-toe », aussi appelé « morpion » ou « X et O » est un jeu simple comportant une grille de 3×3 cases. Chacun des deux joueurs, appelés « X » et « O » joue son tour en inscrivant son symbole dans une case libre. Le premier qui inscrit son symbole sur trois cases alignées, verticalement, horizontalement ou en diagonale gagne la partie. Si les neuf cases sont remplies sans qu'aucun alignement ne soit obtenu, la partie est nulle.

Le joueur O a gagné

```
 1 #include <stdio.h>
 2 char r[3]={' ','X','O'};
 3 void affiche(int t[3][3]){
 4     puts("+---+---+---+");
 5     for(int i=0 ; i<3 ; ++i){
 6         for (int j=0 ; j<3 ; ++j)
 7             printf("| %c ", ..........);
 8         puts("|\n+---+---+---+");
 9     }
10 }
11 ........ joue(int t[3][3], int joueur){
12     printf("Joueur %c :\n",..........);
13     int i, j, k;
14     do {
15         do {
16             printf("Ligne [1-3] ? ");
17             scanf("%d",&i);
18         } while(i<1 || i>3);
19         do {
20             printf("Colonne [1-3] ? ");
21             scanf("%d",&j);
22         } while(j<1 || j>3);
23         k = ..........;
24         if(k!=0)
25             printf("Cette case contient un %c ! Recommencez.\n", .........);
26     } while(.........);
27     t[i][j] = .........;
28     return ((t[i][0]==joueur && t[i][1]==joueur && t[i][2]==joueur)
29           ||(t[0][j]==joueur && t[1][j]==joueur && t[2][j]==joueur)
30           ||(t[0][0]==joueur && t[1][1]==joueur && t[2][2]==joueur)
31           ||(t[2][0]==joueur && t[1][1]==joueur && t[0][2]==joueur));
32 }
33 int main(void){
34     puts("-----------\nTic-Tac-Toe\n-----------\n");
35     int t[3][3] = {{0}};
36     affiche(t);
37     int vainqueur = 0;
38     for(int i=0 ; i<9 && .........; ++i){
39         int joueur = .........;
40         vainqueur = joue(t, joueur)*joueur;
41         affiche(t);
42     }
43     if(.........)
44         printf("Le joueur %c a gagné.\n",r[vainqueur]);
45     else
46         puts("Match nul.");
47 }
```



Exemple d'exécution (fin de partie)

Page annexe 2 : Programme de l'exercice 2 (Décodage)

```
 1 #include<stdio.h>
 2 #include<string.h>
 3 void fill_s(int *s, int m){
 4     s[0]=s[1]=0;
 5     for(int n=2;n*n<m;++n)
 6         if(s[n])
 7             for(int i=n*n;i<m;i+=n)
 8                 s[i]=0;
 9 }
10 void init_s(int s[], int m){
11     while(m--)
12         *s++=1;
13 }
14 void decode1(char *text){
15     int n=strlen(text);
16     int s[n];
17     init_s(s,n);
18     fill_s(s,n);
19     for(int i=0;i<n;++i)
20         if(s[i])
21             putchar(text[i]);
22     puts("");
23 }
24 int ispn(int *sp, int n){
25     for(int j=0;sp[j]*sp[j]<=n;j++)
26         if (n%sp[j]==0)
27             return 0;
28     return 1;
29 }
30 void fill_lp(int *lp, int n){
31     int np=0;
32     lp[np++]=2;
33     for(int i=3;i<n;i+=2)
34         if(ispn(lp,i))
35             lp[np++]=i;
36     lp[np]=0;
37 }
38 void decode2(char *text){
39     int n=strlen(text);
40     int lp[n/2];
41     fill_lp(lp,n);
42     for(int i=0;lp[i]>0 && lp[i]<n;++i)
43         putchar(text[lp[i]]);
44     puts("");
45 }
46 int main(void){
47     char text1[]="Hier attendons cet homme qui n'est pas la";
48     decode1(text1);
49     char text2[]="Mad.brains,tacit.hair.beauty";
50     decode2(text2);
51 }
```