

# Examen #1

## INF3135 groupe 20

Date: Mardi, 21 juin 2005.

Durée: 3 heures.

---

Nom: \_\_\_\_\_  
Code permanent: \_\_\_\_\_

/	5
/	5
/	20
/	15
/	20
/	20
/	15
/	100

---

1. (5 pts) Soit un type de données du nom de **Client**. Laquel des allocations suivantes est correct?
- (a) `Client cl = (Client)malloc (sizeof (Client));`
  - (b) `Client cl = (Client *)malloc (sizeof (Client));`
  - (c) `Client *cl = (Client *)malloc (sizeof (Client));`
  - (d) `Client *cl = (Client)malloc (sizeof (Client));`
  - (e) `Client *cl = (Client)malloc (sizeof (Client *));`
  - (f) `Client *cl = (Client *)malloc (sizeof (Client *));`

---

2. (5 pts) Soit la procédure suivante:

```
void echange(int *x1, int *x2) {
    int t = *x1;
    *x1 = *x2;
    *x2 = t;
}
```

Ajoutez le code dans la procédure **tri** suivante pour appeler la procédure **echange** afin d'échanger le contenu des variables **tab[j]** et **tab[position]**. (L'ajout se fait entre les **\*\*\*\*\***.)

```
void tri(int tab[], int taille) {
    int j;

    while (j < taille - 1) {
        int min = tab[j];
        int i, position = j;

        for (i = j + 1; i++ < taille) {
            if (min > tab[i]) {
                min = tab[i];
                position = i;
            }
        }
        // placer le code pour echanger
        // tab[j] avec tab[position] ici:
        // *****

        // *****
        j++;
    }
}
```

---

3. (20 pts) En langage 'C' les chaînes de caractères sont représentées par un tableau de caractères ayant le caractère '0' à la fin. Une autre façon de représenter une chaîne de caractères est d'utiliser un enregistrement qui contient un tableau de caractères et un entier indiquant la longueur de la chaîne.

(a) (5 pts) Écrivez le code permettant de déclarer (définir) le nouveau type de données **Chaine** décrit ci-dessus.

(b) (15 pts) Écrivez une procédure qui reçoit en entrée une chaîne de type 'C' et qui retourne l'élément de type **Chaine** correspondant.

---

4. (15 pts) Soit les déclarations suivantes:

```
typedef enum {Carre, Cercle} Type;

typedef struct {
    Type sorte;
    union {
        double rayon;
        double cote;
    } Description;
} Formes;
```

Écrivez la procédure dont la signature est:

```
double calculAire (Formes f) {
```

et qui permet de calculer l'aire d'une forme.

Notez:

Aire cercle =  $\pi r^2$

Aire carré =  $c^2$

- 
5. (20 pts) En utilisant la procédure **calculAire** et le type de données **Formes** du numéro précédent. Écrivez le code 'C' qui correspond à l'algorithme de tri (par ordre de surface) suivant:

```
void triInsertion(Formes tableauForme[], int longueurTableau)

pour j de 1 jusqu'a ( longueurTableau - 1 ) faire
    clef = tableauForme[j]
    i = j - 1
    tantque ( i > 0 ) et ( calculAire( tableauForme[i] ) > calculAire( clef ) ) faire
        tableauForme[i + 1] = tableauForme[i]
        i = i - 1
    tableauForme[i + 1] = clef
```

---

6. (20 pts)

Examinez le code suivant:

```
#include <stdio.h>

#define mtM(x) ((x) - 'a' + 'A')
#define itm(x) ((x) - 'A' + 'a')
#define iM(x) ((x >= 'A') && (x <= 'Z'))
#define im(x) ((x >= 'a') && (x <= 'z'))

char s[] = "Bonjour";

int main () {
    int j;

    j = 0;
    while (s[j]) {
        s[j] = iM(s[j]) ? Mtm(s[j]) : (im(s[j]) ? mtM(s[j]) : s[j]);
        j++;
    }

    printf ("%s\n", s);
}
```

(a) (4 pts) Quel message sera affiché par ce programme?

(b) (6 pts) Donnez des noms plus significatifs pour ces identificateurs:

**iM:**

**mtM:**

(c) (10 pts) Décrivez brièvement ce que fait ce code:

---

7. (15 pts) Soit la déclaration suivante:

```
typedef char FonctionEncodage(int, char);
```

Ce type de donnée représente une fonction qui retourne un caractère (char) encodé à l'aide d'une clef d'encodage (int).

Écrivez une procédure C nommée **encoderChaine** qui accepte les entrées suivantes:

- une fonction de calcul d'encodage **FonctionEncodage fe**,
- une clef **int clef**,
- un tableau de caractère **char \*s** (le dernier caractère est un `'\0'`.)

Cette procédure retournera la chaîne de caractère encodée à l'aide de la fonction d'encodage.