

Examen Intra INF3135

Date : Dimanche, 19 mars 2006.

Durée : Trois (3) heures.

Nom: _____
Code permanent: _____

Numéro de groupe et enseignant

20 - Bruno Malenfant

30 - Emmanuel Chieze

/	2
/	6
/	6
/	4
/	3
/	9
/	30

Directives :

- Les documents personnels et les manuels de référence sont autorisés.
- Dans les questions de programmation, vous pouvez faire appel sans réserve aux fonctions de la bibliothèque standard, qui incluent entre autres :
 - `stdio.h` : `sscanf`, `sprintf`, `scanf`, `sprintf`, ...
 - `string.h` : `strcat`, `strcpy`, `strcmp`, `strlen`, ...
 - `stdlib.h` : `malloc`, `free`, ...
 - `ctype.h` : `isalpha`, `isupper`, `toupper`, ...

Numéro 1. (2 pts) Donnez la taille minimale en octets des structures de données suivantes :

a) (0.5 pts) sizeof(struc _forme2D) :

```
struct _forme2D {
    float centre_x;
    float centre_y;
};
```

b) (0.5 pts) sizeof(struct _forme3D) :

```
struct _forme3D {
    struct _forme2D base;
    float centre_z;
};
```

c) (1 pts) sizeof(struct _forme) :

```
struct _forme {
    int nombreDimension;
    char *nom;
    union {
        struct _forme2D forme2;
        struct _forme3D forme3;
    } description;
};
```

Numéro 2. (6 pts) Soit la déclaration suivante, où `struct _forme` est la structure déclarée au numéro précédent :

```
typedef struct _forme Forme;
```

Écrivez le code d'une fonction qui alloue et construit un élément de type `Forme` et ce pour une forme à trois (3) dimensions. Cette fonction aura donc l'en-tête suivant :

```
Forme *creerForme3D( float centre_x, float centre_y, float centre_z, char *nom );
```

Numéro 3. (6 pts) Un coefficient binomial est calculé de la façon suivante :

$$C(n, k) = C(n-1, k-1) + C(n-1, k)$$

$$C(n, 0) = 1$$

$$C(n, n) = 1$$

Écrivez le code 'C' qui réalise cette fonction récursive.

Numéro 4. (4 pts) Indiquez ce qui sera affiché par le programme suivant :

```
#include <stdio.h>

#define NB 5

int main() {
    int i = 0, j = 0;
    int tab[NB][NB] = { { 1, 1, 1, 1, 1 },
                        { 0, 0, 0, 0, 0 },
                        { 0, 0, 0, 0, 0 },
                        { 0, 0, 0, 0, 0 },
                        { 0, 0, 0, 0, 0 } };

    for( i = 1; i < NB; i++ ) {
        for( j = 1; j < NB; j++ ) {
            tab[i][j] = tab[i][j-1] + tab[i-1][j-1];
        }
    }

    for( i = 0; i < NB; i++ ) {
        for( j = 0; j < NB; j++ ) {
            printf( "%i", tab[i][j] );
            if ( j < NB-1 ) { printf( " " ); }
            else { printf( "\n" ); }
        }
    }

    return 0;
}
```

Numéro 5. (3 pts) Indiquez ce qui sera affiché par le programme suivant :

```
#include <stdio.h>

int p = 2;
int f( int q ) {
    return p = q++;
}

int main() {
    int r[15] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};
    int p = 5;
    int q = 7;
    int i = 0;

    r[i++] = p + q;
    {
        int p = --q;
        r[i++] = p + q;
    }

    q += i++;
    r[++i] = p;

    r[i++] = f( q );
    r[i++] = f( q );

    for( ; i >= 0 ; i-- ) {
        printf( "%i,", *(r+i) );
        if( i == 0 ) { printf( "\n" ); }
        else          { printf( ", " ); }
    }

    return 0;
}
```

Numéro 6. (9 pts) Soit les quatres tableaux suivants qui contiennent chacun 100 chaînes de caractères :

```
char *article[100];  
char *nom[100];  
char *verbe[100];  
char *preposition[100];
```

a) (6 pts) On suppose que ces tableaux sont des variables *globales* et qu'aucun mot contenu dans ces tableaux n'aura plus de 20 caractères.

Vous devez coder une fonction `genererPhrase` qui va construire une phrase aléatoire générée selon un descripteur, fonction qui aura l'en-tête suivant :

```
char *genererPhrase( char *descripteur )
```

Un descripteur de phrase est une chaîne de caractères (donc un tableau de caractères), où chaque caractère est une des lettres 'a', 'n', 'v', 'p' et où le tableau se termine par le caractère ASCII nul ('\0'). Chaque caractère indique un certain type de mot qui devra être généré pour la phrase.

Par exemple, "anvpan" est un descripteur qui indique une phrase qui doit commencer par un (a)rticle suivi d'un (n)om, d'un (v)erbe, d'une (p)roposition, d'un (a)rticle et finalement d'un (n)om.

Votre fonction `genererPhrase` doit donc recevoir en entrée un tel descripteur. Elle va alors construire une chaîne de caractères contenant une phrase dont la forme respecte le descripteur fourni. Cette phrase devra commencer par une majuscule et se terminer par un point.

Le générateur aléatoire suivant vous est fourni :

```
int aleatoire( int max );  
// Retourne une valeur aléatoire entière comprise entre 0 et max-1 (inclusivement).
```

b) (3 pts) Écrivez le code d'un programme principal qui reçoit en argument une chaîne contenant un descripteur, qui appelle la fonction `genererPhrase`, puis qui affiche le résultat sur `stdout`.