

Devoir Surveillé Algorithmique et programmation C II (2017/2018)

1ère année

—Enseignante : Mme Aïcha El Golli Jabbes—

1h30 – Documents et calculatrices interdits

► Exercice 1. Appel Fonction et procédure Récursive (3 Points)

Fonction Devine(n :entier) :entier

Début

si(n=0)ou(n=1) alors

retourne(n);

sinon

retourne Devine(n-2)+Devine(n-1);

FinSi;

Fin

1. Quelle est la valeur retournée par l'appel de Devine(4) ?
2. Traduire la fonction Devine() en langage C

► Exercice 2. Les chaînes de caractères et pointeurs (6 Points)

1. Soit le code suivant :

```
char ch1[] = "";  
char ch2[] = "ab";  
char ch3[] = {'a', 'b'};  
char *ch4=(ch3+1);  
*ch4='i';
```

- (a) Que contient ch1[0] ?
 - (b) que retourne strlen(ch2) ?
 - (c) que contient ch3[2] ?
 - (d) que contient le tableau ch3 ?
 - (e) est ce que ch3 est une chaîne de caractère ?
2. Soit le code suivant :

```
void g(int *p, int j) { *p +=j; j+=*p; }  
void main(){  
    int *p, i=5, j=3;  
    p=&i;  
    g(p, j);  
    printf("%d %d %d\n", *p, i, j); }
```

Qu'est ce qui s'affiche ?

► Exercice 3. Les structures (Le compte en banque - 11 points)

Le programme informatique d'une banque, repose sur la structure suivante pour représenter les comptes de ses clients.

```
typedef struct com{
    int num; /* numero du compte */
    char nom[30]; /* nom du proprietaire du compte */
    char adresse[100]; /* adresse du proprietaire */
    float solde; /* le solde du compte au debut du mois en cours */
    operations op; /* les operations du mois en cours */
} compte;
```

Le champ op de la structure permet d'enregistrer les opérations en cours de mois afin de calculer le nouveau solde du compte lorsque l'on change de mois.

La structure du type ope permet d'enregistrer une opération et qui est donc un élément de structure operations. Chaque opération contiendra les champs suivants :

- libelle : une chaîne contenant le libellé de l'opération
- type : un caractère, @ si l'opération est un débit, et C si il s'agit d'un crédit.
- montant : le montant de l'opération un nombre réel positif qui sera à ajouter au solde si l'opération est un crédit et à retrancher si il s'agit d'un débit.

Le type operations qui n'est autre qu'une structure composée d'un tableau t de type ope (max 200 opérations) et une taille déterminant le nombre d'opérations contenu dans t.

```
typedef struct {
    char libelle[20];
    char type;
    float montant;
} ope;
typedef struct{
    ope t[200];
    int taille;
} operations;
```

1. Écrire une fonction récursive SommeOperation qui prend en paramètre un compte et qui renvoie la somme des opérations contenus dans op. N'oubliez pas qu'une opération débit c'est un retrait (-) et une opération crédit est un dépôt d'argent (+).
2. Donnez la définition de la fonction NouveauSolde qui prend en paramètre un compte et qui renvoie le nouveau solde du compte en fonction des opérations qu'il contient (utilisant la fonction précédente)
3. Définir la fonction NouvelleOperation qui prend en paramètre toutes les informations d'une opération et renvoie un ope dont les différents champs ont été correctement affectés. Vous pouvez utiliser la bibliothèque *string.h*.
4. En utilisant la fonction précédente écrire la procédure AjouteOperation qui prend en paramètre un pointeur sur un compte et toutes les informations d'une opération et qui l'ajoute aux opérations dans le tableau des opérations du compte.

Bon travail

Devoir Surveillé Algorithmique et programmation C II (2017/2018)

1ère année

—Enseignante : Mme Aïcha El Golli Jabbes—

1h30 – Documents et calculatrices interdits

► Exercice 1. Appel Fonction et procédure Récursive (3 Points)

Fonction Devine(n :entier) :entier
Début
si(n=0)ou(n=1) alors
retourne(n);
sinon
retourne Devine(n-2)+Devine(n-1);
FinSi;
Fin

```
int Devine (int n)//(0.25pt)
{
    if ((n==0) || (n==1))//(0.5pt)
        return n;//(0.25pt)
    else
        return (Devine(n-2)+Devine(n-1)); //(0.5pt)
}
```

1. Quelle valeur est retournée par l'appel de Devine(4) ?
2. Traduire la fonction Devine() en langage C

► Exercice 2. Les chaînes de caractères et pointeurs (6 Points)

1. Soit le code suivant :

```
char ch1[] = "";
char ch2[] = "ab";
char ch3[] = {'a', 'b'};
char *ch4=(ch3+1);
*ch4='i';
```

- (a) Que contient ch1[0] ? le caractère de fin de chaîne (1pt)
- (b) que retourne strlen(ch2) ? 2 (0,5pt)
- (c) que contient ch3[2] ? n'existe pas (1pt)
- (d) que contient le tableau ch3 ? le tableau de caractère constitué de 'a' et 'i' (0,5pt)
- (e) est ce que ch3 est une chaîne de caractère ? non (1pt)

2. Soit le code suivant :

```
void g(int *p,int j) { *p +=j; j+=*p;}
void main(){
    int *p, i=5, j=3;
    p=&i;
    g(p,j);
    printf("%d %d %d\n", *p,i,j); }
```

Qu'est ce qui s'affiche ? 8 8 3 (2pt)

► Exercice 3. Les structures (Le compte en banque - 11 points)

Le programme informatique d'une banque, repose sur la structure suivante pour représenter les comptes de ses clients.

```
typedef struct com{
    int num; /* numero du compte */
    char nom[30]; /* nom du proprietaire du compte */
    char adresse[100]; /* adresse du proprietaire */
    float solde; /* le solde du compte au debut du mois en cours */
    operations op; /* les operations du mois en cours */
} compte;
```

Le champ op de la structure permet d'enregistrer les opérations en cours de mois afin de calculer le nouveau solde du compte lorsque l'on change de mois.

La structure du type ope permet d'enregistrer une opération et qui est donc un élément de structure operations. Chaque opération contiendra les champs suivants :

- libelle : une chaîne contenant le libellé de l'opération
- type : un caractère, d si l'opération est un débit, et c si il s'agit d'un crédit.
- montant : le montant de l'opération un nombre réel positif qui sera à ajouter au solde si l'opération est un crédit et à retrancher si il s'agit d'un débit.

Le type operations qui n'est autre qu'une structure composé d'un tableau t de type ope (max 200 opérations) et une taille déterminant le nombre d'opérations contenu dans t.

```
typedef struct {
    char libelle[20];
    char type ;
    float montant;
} ope;
typedef struct{
    ope t[200];
    int taille;
} operations;
```

1. Écrire une fonction récursive SommeOperation qui prend en paramètre un compte et qui renvoie la somme des opérations contenus dans op. N'oubliez pas qu'une opération débit c'est un retrait (-) et une opération crédit est un dépôt d'argent (+).

```
float SommeOperation(compte c)//4pt
{
    if(c.op.taille==0) return 0.0;
    else
    {
        c.op.taille--;
        if(c.op.t[c.op.taille].type=='c')
            return(c.op.t[c.op.taille].montant+SommeOperation(c));
        else
            return(-c.op.t[c.op.taille].montant+SommeOperation(c));
    }
}
```

2. Donnez la définition de la fonction NouveauSolde qui prend en paramètre un compte et qui renvoie le nouveau solde du compte en fonction des opérations qu'il contient (utilisant la fonction précédente).

```
float NouveauSolde(compte c){//2pt
    return (c.solde+SommeOperation(c));
}
```

3. Définir la fonction `NouvelleOperation` qui prend en paramètre toutes les informations d'une opération et renvoie un `ope` dont les différents champs ont été correctement affectés. Vous pouvez utiliser la bibliothèque `string.h`.

```
ope NouvelleOperation(char *lib, char type, float mt){ //3pt
ope op;
strcpy(op.libelle, lib);
op.type=type;
op.montant=mt;
printf("%f\n",op.montant);
return op;
}
```

4. En utilisant la fonction précédente écrire la procédure `AjouteOperation` qui prend en paramètre un pointeur sur un compte et toutes les informations d'une opération et qui l'ajoute aux opérations dans le tableau des opérations du compte.

```
void AjouteOperation(compte *c, char *lib, char type, float mt){ //2pt
ope o= NouvelleOperation(lib,type,mt);
c->op.t[c->op.taille]=o;
c->op.taille++;
}
```

Bon travail