EMSA

Langage C — Avril 2017

Durée : 1H30 - Documents et Calculatrices interdits — A. CHAHBOUN

Question I : (4 points)

1- Expliquez la notion de symbole ?
2- Expliquez la notion de post-incrémentation et Pré-incrémentation ?
3- En cas de désaccord entre le code format et le type de l'expression dans une fonction printf, que fait le compilateur ?
4- Expliquez la notion de tampon et de séparateurs lors de l'utilisation de la fonction scanf ?
5- Lors du traitement d'un code de format, dans quelles conditions la fonction scanf arrête l'exploration du tampon ?
6- Quelles sont les valeurs de retour de la fonction printf et scanf ?
7- Combien d'instruction return peut-on avoir dans un programme ?
8- Dans les tableaux, que fait le compilateur pour traiter le débordement d'indices ?

Question II : Operateurs et Formats (4 points)

```
A=20 B=5 C=-10 D=2 X=12 Y=15
1) A == (C=5)
2) A != (C *= (-D))
3) A %= D++
4) A %= ++D
5) (X++)*(A+C)
6) A = X*(B<C)+Y*!(B<C)
7) !(X-D+C)||D+A
8) ((A&&B)||(!0&&C))&&!D
9) ((A&&B)||!0)&&(C&&(!D))
10) C+=A>0&&A<=15?++A:A/B;
```

```
#include <stdio.h>
main ()
{ int n = 123 ;
  int p = 8 ;
  float x = 33.5566;
  printf("1 : %d %f\n", n, x) ;
  printf("2 : %5d %12e\n", n, x) ;
  printf("3 : %2d %3e\n", n, x) ;
  printf("4 : %11.1f %10.2e\n", x, x) ;
  printf("5 : %e\n", x) ;
  printf("6 : %*.*f\n", 9, 1, x) ;
}
```

Question III : Fonctions Récursives (4 points)

Définir la fonction récursive qui calcule la somme des éléments d'un tableau TAB déclaré de manière globale ?

```
int TAB[10], n;
int somme (int n)
{
      //instructions à réaliser.
}
```

Question IV : Tableaux (4 points)

Soit le tableau d'entiers suivant : [103 117 128 137 136 52 53 20]

1. Définir la fonction void afficher_pair() qui affiche seulement les nombres pairs du tableau. Exemple d'affichage : message[5]=52.
2. Définir la fonction void decoder_message(int x) qui soustrait la valeur x à chacun des entiers du tableau.
3. Ecrire une fonction main() qui dans l'ordre :
   a. affiche les entiers pairs du tableau
   b. soustrait 20 aux entiers du tableau
4. Supposant que le tableau contient des valeurs de résistances, définir une fonction qui retourne la résistance équivalente d'un montage parallèle.

Question V : Instructions de contrôle (4 points)

Ecrire un programme qui trouve la plus grande et la plus petite valeur d'une succession de notes (nombres entiers entre 0 et 20) fournies en données, ainsi que le nombre de fois où ce maximum et ce minimum ont été attribués. On supposera que les notes, en nombre non connu à l'avance, seront terminées par une valeur négative (ne pas utiliser de "tableau").

CC 2016/2017 langage C

Question I :

1 - la Notion de symbole, en C, symbole est un caractère ou ensemble de caractère regroupé, cette notion utilisé lors on donne un valeur à un mots (symbole), dans la zone directive (#define).

2 - post incrementation : incrementation du variable se fait tout les opérations.
pré incrementation : l'incrementation se fait en premier, avant tout les operations.

3 - En cas de désaccord entre le code format et le type de l'expression dans la fonction printf, si ils ont la même taille, les conséquences se limitent pour une mauvaise interprétation, s'ils ont un taille déférents, ~~les mauvaise conséquences~~ perdre de precesion, decalage du valeur, ...

4 - la mémoire tampon est une emplacement memoire temporelle, où les valeurs saisir dans le clavier sont stocké ~~avant la rencontre~~, lors du saisi, le tampon faire avance tout les caractère separateurs jusqu'à première caractère different, (~~es~~ sauf type char ou s); et il prend tous les valeurs jusqu'à la rencontre du premier charactère separateur.

un separateur sert à arreter la saisie et à scanf de prendre tout les valeur saisi avant.

5 - - Gabarit maximal
- caractère invalide.
- Separateur (espace, retour à la ligne).

6 - Scanf et printf retourne respectivement le nombre de caractère saisi, et le nombre de caractère à afficher

7 - on peut avoir ~~n'importe~~ plusieur instruction return dans un programme, ou bien aucune return, mais le point descaté est que return interompt l'execution du programme.

8 - dans C, il n'y a aucun control à propos de debordement d'indice dans un tableau, et le compilateur prend n'importe qu'elle valeur ~~dans~~ ~~la-ta~~ lors du debordement.

Question II : (on va les considérés int)

1 - 0
2 - 0
3 - 10
4 - 10
5 - 1?0

6 - 15
7 - 1
8 - 1
9 - 0
10 - 1

1 - 1?3
2 - ^^1?3

33.556600
3.355660e+001

3- 123 | 3.35566+e001 ; (3)

4- ^^^^^^33.5 | ^3.35e+001 ;

5- 3.355660e+001

6- ^^^^^33.5

Question III :

calcule la somme des elts du TAB déclaré de manière globale.

```
int TAB[10], n;        // n dimension du tableau
int somme (int n){
  if (n>=1){ return ( T[n-1] + somme(n-1)); }
  else { return (T[n]); }
}
```

Question IV :

1- fonction Void affiche seulement les nombre pair du tableau.

```
Void affiche_pair (int T[8], ...){
  int i;
  for (i=0; i<8...; i++){
    if (T[i]%2==0)
      printf("message[%d] = %d \n", i, T[i]);
  }
}
```

2 - Void decode_message (int x){
int i;
for(i=0; i<8; i++){
T[i] = T[i] - x;}
}.

3 - programme principale :

```
int T[8];
main(){
  int i;
  for (i=0; i<8; i++){
    printf ("saisir la valeur %d", i+1); // saisir
    scanf("%d", &T[i]);}
  affiche_pair(T);
  decode_message(20);
}
```

4 - Resistance equivalent d'un montage //.

$\frac{1}{R_{eq}} = \sum \frac{1}{R_i}$.

```
float R_eq(int T[8]){
  float R=0;
  for(i=0; i<8; i++){
    R = R + (float) 1/T[i];}
  return (1/R);}
```

```c
#include<stdio.h>
int T[10];
int somme(int n){
int res;
if (n>=10){
    return -1;
}
else {
    if(n==0){
        res=T[0];
        return res;
    }
    else{
        res=T[n]+somme(n-1);
        return res;
    }
}
}
```

```
3   void afficher_pair(){
4   int i;
5   for(i=0;i<8;i++){
6       if(T[i]%2==0){
7           printf("message[%d]=%d \n",i,T[i]);
8       }
9
10  }
11  printf("\n");
12  }
13
14  void decoder_message(int x){
15  int i;
16  for(i=0;i<8;i++){
17      T[i]=T[i]-x;
18      }
19  }
20
21  int main(){
22  void afficher_pair();
23  void decoder_message(int x);
24
25  afficher_pair();
26  decoder_message(20);
27
28  return 0;
29  }
30
```

```c
#include<stdio.h>
int main(){
int j=1,k=1,min=20,max,n;
printf("saisir les notes entre 0 et 20 : \n");
scanf("%d",&n);
if(n<0){
    printf("votre list est vide\n");
}
else {
max=n;
while(n>0) {
    while(n>20){
        printf("saisir une note valide (entre 0 et 20) : ");
        scanf("%d",&n);
    }
    if(n<0) break;

    if (n>max){
        max=n;
        j=1;}
    else if (n==max){
        j++;
    }


    if(n<min){
            min=n;
            k=1;
        }
    else if (n==min){
            k++;
        }

scanf("%d",&n);
}

```