QROC RAN Informatique – Décembre 2011

Répondre sur la feuille impérativement. Notes personnelles et polycopiés distribués non autorisés.

NOM : Correction JF COLIN
Prénom :
Filière :

Compilation

⑩ Chaque ligne du tableau ci-dessous contient une portion de code qui doit être considérée comme indépendante des autres. **Indiquer (par une croix) si la portion de code est correcte ou provoque une erreur de compilation ou provoque une erreur du *linker* ou peut provoquer un effet indésirable à l'exécution**

Code	Correct	Pb Compilation	Pb Linker	Pb possible à l'exécution

`void trb(int m);`
`int main() {`
`  int n = 7;`
`  n = n * n;`
`  trb(n + 5);`
`  return 0 ;`
`}`
			X	

`int main() {`
`  double x = 7;`
`  y = x + 2;`
`  printf("x=%lg y=%lg\n", x, y);`
`  return 0 ;`
`}`
		X		

`int t[10];`
`int main() {`
`  int n = 7;`
`  for (int i = 0; i < 10; i++)`
`  t[i] = n * n;`
`  for (int i = 0; i < 10; i++)`
`  printf("%d", t[i]);`
`  return 0;`
`}`
	X			

`int main() {`
` int i,n=3,m=0;`
` for(i=0;i<n;i++) {`
`  scanf("%d",n);`
`  if (n>m) m=n;`
` }`
` printf("max = %d\n",m);`
` return 0;`
				X

```
}
```

Fonctions-procédures

⑩ *Compléter le code ci-dessous*

Remarque : l'exercice peut être fait sans connaître l'indication suivante :
le code ASCII de A vaut 65, le code ASCII de a vaut 97

```c
#include <stdio.h>
#include <stdlib.h>

int isLower(char c) {

    return c>='a' && c<='z' ;

}

int isUpper(char c) {

    return c>='A' && c<='Z' ;

}


int main(void) {
       char c='5';
       if (isLower(c)) printf ("%c est en minuscule ",c);
       else if (isUpper(c)) printf("%c est une majuscule",c);
       else printf("%c est un symbole de ponctuation ou un chiffre",c);
       return EXIT_SUCCESS;
}
```

Procédure

On considère le programme suivant qui reprend les deux fonctions précédentes (et qui suppose qu'elles ont été correctement écrites...). Il est conseillé de réutiliser ces deux fonctions dans ce qui est demandé ci-dessous..
Un exemple d'exécution de ce programme est le suivant :

```
Entrer une phrase : Salut Les Gars !!!
SALUT LES GARS !!!
salut les gars !!!
```

⑩ **Compléter les deux procédures** toLower et toUpper en conséquence.

```c
int isLower(char c) {
       // code non dévoilé... , mais supposé correct !
}

int isUpper(char c) {
```

```
       // code non dévoilé... , mais supposé correct !
}

void toLower(char *s, char *t) {
       int lg=strlen(s),i ;
      for(i=0;i<lg;i++)
         if (isUpper(s[i]) t[i]=s[i]+'a'-'A' ;
         else t[i]=s[i] ;
      t[i]=0 ;

}

void toUpper(char *s, char *t) {
       int lg=strlen(s),i ;
      for(i=0;i<lg;i++)
         if (isLower(s[i]) t[i]=s[i]+'A'-'a' ;
         else t[i]=s[i] ;
      t[i]=0 ;

}

char phrase[1000];
char t[2000];

int main(void) {
       printf("Entrer une phrase : ");
       gets(phrase);
       toUpper(phrase,t);
       printf("%s\n",t);
       toLower(phrase,t);
       printf("%s\n",t);
       return EXIT_SUCCESS;
}
```

Langage C

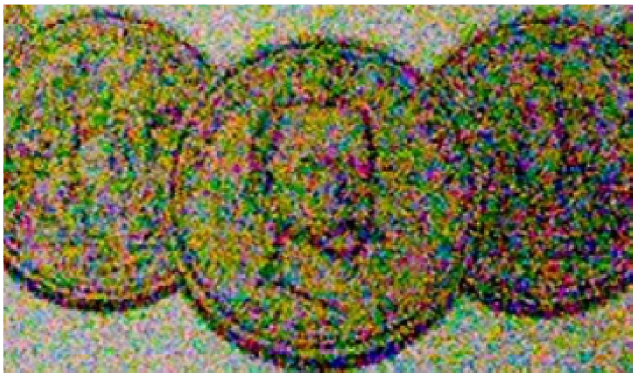
Un capteur numérique d'appareil photo est assimilé, dans le logiciel embarqué, à 3 tableaux à deux dimensions qui gèrent respectivement la couleur rouge, verte et bleue du signal.

```
#define RESOX 4912
#define RESOY 3264

int capteurRed[RESOX][RESOY] ;
int capteurGreen[RESOX][RESOY] ;
int capteurBlue[RESOX][RESOY] ;
```

L'utilisation d'un tableau à deux dimensions en C est très simple : l'accès à l'élément de coordonnées ***i*** et ***j*** s'exprime par, par exemple, ***capteurRed[i][j]*.**

Quand l'image est prise dans des conditions de faible lumière le signal délivré sur chaque pixel du capteur est très amplifié et cela engendre une détérioration du rapport signal/bruit . Concrètement cela se traduira par l'existence de pixels aléatoires anormalement brillant ou sombre dans chaque couleur.

	
Image peu bruitée	Image très bruitée

Pour atténuer l'effet de ce bruit l'algorithme embarqué dans l'appareil photo met en œuvre sur l'image très bruitée un premier traitement de lissage qui remplace chaque valeur de pixel repérée comme anormalement basse ou élevée par rapport au voisinage par la valeur moyenne de ses voisins immédiats.
Par exemple l'appel suivant, qui n'agit que sur le tableau rouge:
m= moyenneRed(127,351) ;
a pour effet d'affecter *m* avec la valeur moyenne des pixels adjacents au point (127,351) du capteur rouge.

	125	*126*	*127*	*128*	*129*	*130*
348	29	20	1	17	43	17
349	16	4	36	9	16	49
350	34	**36**	**2**	**18**	25	15
351	6	**32**	**187**	**27**	19	17
352	15	**4**	**7**	**31**	6	37
353	38	28	16	8	29	32

Extrait du tableau rouge

L'appel *moyenneRed(127,351)* retourne la valeur moyenne de 36,2,18,27,31,7,4,32 soit 20 (valeur arrondie)

⑩ **Écrire la fonction *moyenneRed (sur copie distribuée)***

```
int moyenneRed(int line, int  col) {
        int i,j ;
        int sum=0, count=0
        for(i=line-1;i<=line+1;i++)
                for(j=col-1;j<=col+1;j++) {
                        if (i>=0 && i<RESOX && i!=j && j>=0 && j<RESOY ) {
                                sum+=capteurRed[i][j] ;
                                count++ ;
                        }
        }
        return sum/count ;
}
```