



UNIVERSITÉ PARIS DESCARTES

IUT

DÉPARTEMENT INFORMATIQUE

DISCIPLINE : TAP

Date de l'épreuve : 7/11/17

Année : 2017-18 Groupe : 109

NOM : AROSTOUCET
 Prénom : ANNE
 Ecrire très lisiblement
 (en capitales)

NOTE DE 0 À 20

16,5/20

APPRÉCIATIONS

KE Très bon travail, à encourager!

Ne rien écrire dans cette marge

Partie I

6,75
10,5

6
10,5

Très Bien

1) a)

d) → ? 0

4 6
5 6

oui
Très bien

e) → 0,5

b)

96 oui

~~y = 10~~

non

f)

c)

~~val = 3 * 7 = 2~~ 0

2 3 ✓

2 3 ✓

5 5 ✓

2 5 ✓

Très Bien

g)

36

6 → 2

6

0,5

1/10

2)
 * [int] Produkt p, m ∈ ℕ
 * [int] : a ∈ ℕ
 * [int] : b ∈ ℤ
 * [return] : BigO

0,5

void ~~float~~
 Big h (Produkt p, unsigned int m, int⁺ b);

0,5

```
int main {
  Produkt a;
  unsigned int b = 17;
  int+ c = 4;
  Big res = h(a, b, &c);
}
```

0,75

```
unsigned int i;
3) unsigned char f;
for (i = 1; i < 5; ++i) {
  for (j = 1; j < 5; ++j) {
    printf("%d", f);
  }
}
```

0,5
0,75

```

4) /* my strcpy
   * [in-out] src, dest
   * [in] src-taille
   void my strcpy (char* src, char* dest,
   unsigned int src-taille) {
   assert (src-taille <= 255);
   unsigned int i;
   for (i=0; i < src-taille+1; ++i) {
   *dest[i] = *src[i]
   }
   }

```

Comme copie pas!

0,75
1

```

int main {
char source[255];
scanf ("%s", source);
char copie[255];
my strcpy (source, copie, strlen(source));
printf ("source = %s\n", source);
printf ("copie = %s\n", copie);
system ("pause"); return 0;
}

```

↑ source importante
= : et
= #

0,75

```

5) a) typedef struct {
int nombre_dif;
char pays_dif[70];
double prix;
char anhet-patit[20];
Image2D image;
} Timbre

```

me

0,5
0,75

```

b) typedef struct {
timbre collec[1000];
} Collection

```

```

double px = maCollection.collec[0].prix;
float

```

d) `imageID im = la(collection) -> collection -> image;`

Partie II

1) a)

Les résultats attendus sont:

(i) 3 et 2 ? } min 3

(ii) 7 et 6 ? } max 4

b) ? pas d'aire } min 2

0,5

`/* minMaxOp`

`* [in] : i, j, tab, size`

`* [out] : min, max, moyenne`

`* Typed of struct } } ;`

0,75

`void minMaxOp(unsigned int i, unsigned int j, int tab, int* min, int* max, float* moyenne, unsigned int size);`

c)

`/* minMaxOp`

`* [in] : i, j, tab, size`

`* [out] : min, max, moyenne`

`* Pré-conditions :`

`* size > 0`

`* i < j`

`*`



UNIVERSITÉ
PARIS
DESCARTES

IUT

DÉPARTEMENT INFORMATIQUE

DISCIPLINE : IAD

Date de l'épreuve : 7/11/17

Année : 2016-17 Groupe : 102

NOM : CAROSTELET
Prénom : Alexandre
Écrire très lisiblement
(en capitales)

NOTE DE 0 À 20

APPRECIATIONS

Très Bon Travail
Mauvaise structure, mauvaise organisation - Écrire lisible!

Ne rien écrire dans cette marge

d) 10,25/17,5 → 10/17,5

```

void minMaxOf(unsigned int i, unsigned
int j, int tab[], int *min, int *max,
float *moyenne, unsigned int size) {
    assert(size > 0);
    assert(i < j);
    unsigned int cpt;
    *min = 1000; *max = -1000;
    for (cpt = i; cpt < j; ++cpt) {
        if (*min > tab[cpt])
            *min = cpt;
    }
    for (cpt = i; cpt < j; ++cpt) {
        if (*max < tab[cpt])
            *max = cpt;
    }
    int sum = 0;
    for (cpt = i; cpt < j; ++cpt) {
        sum += tab[cpt];
    }
}
    
```

1,95

5/10
UMU

* moyenne = sum / cpt;

e) /* "minMaxOf.c" Auteur (APRISTOLET 7/11/19)

#include <assert.h>

#include <stdio.h>

int main() {

int tab = { 3, 25, 2, -10, 15, 5, -1, 100 }; // (1)

unsigned int size = sizeof(tab) / sizeof(int);

* test unitaire 1 * // (3)

unsigned int i = 2, j = 6;

int min, max;

float moyenne;

minMaxOf(i, j, tab, &min, &max, &moyenne, size);

printf("min = %d", min);

printf("max = %d", max);

printf("moyenne = %f", moyenne);

assertion

/* test unitaire 2 * / scanf(" ");

i = 3; j = 100;

minMaxOf(i, j, tab, &min, &max, &moyenne, size);

printf("min = %d", min);

printf("max = %d", max);

printf("moyenne = %f", moyenne);

assertion

```
/* Autre mode */ // (4)
```

```
unsigned int cpt;  
for (cpt=0; cpt<10; ++cpt) {  
    scanf ("%d %d" &i, &j);  
    minMaxOf (i, j, tab, &min, &max,  
    &moyenne, size);  
    printf ("i=%d, j=%d, moyenne=%d,  
    indice du min=%d, indice du max=%  
    %d.", i, j, moyenne, min, max);  
}
```

```
2) a) /* pileibniz:
```

```
* [in] : n nombre d'iterations  
* [return] : approximation de pi  
* math.h est requis */
```

```
double pileibniz (unsigned int n); ✓
```

b)

```
double pileibniz (unsigned int n);
```

```
double pi-approx;
```

```
unsigned int i=0;
```

```
for (i=1; i<n; ++i) { ✓
```

```
    pi-approx += (pow(-1, i) / ((2*i+1)));  
}
```

```
pi-approx = (1 - pi-approx) * h; ✓
```

```
return pi-approx;
```

0,5

1,5

/*"pileipmiz.c" Arsène CAPOSTOLET 7/11/17*/

```
#include <math.h>
#include <stdio.h>
```

0/5

```
int main() {
    unsigned int n = 1000000;
    printf("L'estimation de pi avec
    la formule de Leibniz a 10E-6
    près est %f", pileipmiz(n));
    system("pause"); return 0;
}
```

```
1) /* Comptine Numerique.c */
/* Arsène CAPOSTOLET
* 7/11/17
*/
```

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

3/5

```
2) /* prototypes des fonctions */
```

```
void hasard (unsigned int m, unsigned
int *k, unsigned int *p);
unsigned int comptine (unsigned int k,
unsigned int p);
```

```
int main() {
    unsigned int m, k, p;
    scanf ("%d", &m);
    hasard (m, &k, &p);
    printf (" %u participant, nombre
de la comptine: %u, \n", m, k);
    printf (" numéro du participant
du début de la comptine: %d \n",
p);
}
```

0/10



UNIVERSITÉ PARIS DESCARTES

IUT

DÉPARTEMENT INFORMATIQUE

DISCIPLINE :

IAP

Date de l'épreuve :

7/11/17

Année :

2017-18

Groupe :

109

Ecrire très lisiblement

NOM : LABOSTOLET
(en capitales)

Prénom : Assène

NOTE DE 0 À 20

APPRECIATIONS

Ne rien écrire dans
cette marge

```
printf("Le gagnant est %d !!!\n",  
comptime(k, p, m);  
system("pause"); return 0; ✓
```

/ définition des fonctions */*

/ Hasard*

** [in] : m*

** [out] : k, p*

** les conditions :*

** $0 < k < m$*

** $m > 0$*

** $1 < p < m$*

** ✓*

```
void hasard (unsigned int m, unsigned  
int * k, unsigned int * p) {  
assert (m > 0)
```

```
✓ srand ((unsigned)time (NULL)); over
```

** k = rand () % m + 1;*

assert (k < m && k > 0);

** p = rand () % m + 1;*

~~assert (p < m && p > 0);~~

9/10

```

/* Comptine
 * [int] k, p, m
 * [return] gapmant
 */

```

```

unsigned int Comptine (unsigned int k,
unsigned int p, unsigned int m) {
    unsigned int gapmant;
    unsigned int i = p;
    unsigned int tab[m];
    for (i=0; i < m; ++i) {
        tab[i] = 0 1;
    }
    for (i=0; i < m-1; ++i) {
        for (j=0; j < k; ++j) {
            j += 1;
        }
        j = j % m;
        tab[i] = j;
    }
    for (i=0; i < m; ++i) {
        if (tab[i] == 0) {
            gapmant = tab[i];
        }
    }
    return gapmant;
}

```

Bonus +0,25