+1/1/60+

Nom, prénom, numéro d'étudiant : .....

MPI, Info 111, Partiel du 28 octobre 2013 (deux heures)

Tous documents autorisés, sauf les annales.
Les réponses sont à donner sur le sujet. Les réponses aux questions à choix multiples doivent être données sur la dernière page. Ces questions ont une unique bonne réponse. Des points négatifs sont affectés en cas de mauvaise réponse.

Exercice 1 (Questions à choix multiples)**.**
Qu'affichent chacun des (fragments de) programmes suivants ?

Question 1

```
int g(int x, int y) {
    x = x+1;
    return y-1;
}

int main() {
    int x,y;
    x = 1;
    y = 2;
    g(x,y);
    cout << x << ", " << y << endl;
}
```

A $1, 2$ B $1, 1$ C *Erreur* D $2, 1$

Question 2

```
    int x, y;
    x = 0;
    y = 0;
    while ( x < 10 ) {
        x = x + 1;
        y = 1;
    }
    cout << x << ", " << y << endl;
```

A $9, 9$ B $0, 1$ C $9, 1$ D $10, 1$

Question 3

```
    int x, y;
    x = 5*(2/5);
    y = (5*2)/5;
    cout << x << ", " << y << endl;
```

A $2, 0$ B $0, 0$ C $2, 2$ D $0, 2$

Pour votre examen, imprimez de préférence les documents compilés à l'aide de auto-multiple-choice.

Question 4

```
int f(int x, int y) {
    x = x+1;
    return y-1;
}

int main() {
    int x,y;
    x = 1;
    y = 2;
    x = f(y,x);
    cout << x << ", " << y << endl;
}
```

A $0, 2$ B $1, 2$ C $1, 1$ D $2, 1$

Question 5

```
    int x, y;
    y = -3;
    x = -y;
    if ( y > 0 ) {
        y = 2*x;
    } else {
        y = -y;
    }
    cout << x << ", " << y << endl;
```

A $-3, -3$ B $3, 3$ C $6, 3$ D $6, -3$

Question 6

```
void f(int x, int y) {
    x = x+1;
    y = y-1;
}

int main() {
    int x,y;
    x = 1;
    y = 2;
    f(y,x);
    cout << x << ", " << y << endl;
}
```

A $3, 0$ B *Erreur* C $2, 1$ D $1, 2$

Exercice 2 (Fonctions)**.**

1. Implanter une fonction itérative `puissanceIteratif(n, k)` qui reçoit deux entiers positifs en paramètres et renvoie n^k.

2. Implanter une fonction récursive `puissanceRecursif(n, k)` ayant le même comportement.

Exercice 3 (Jeu du pendu)**.**
On souhaite implanter le jeu du pendu, selon le principe suivant : un mot est choisi par le programme au début du jeu. Au départ, il est affiché avec toutes les lettres masquées par des étoiles. À chaque tour, le joueur choisit une lettre entre 'a' et 'z'. Chaque occurrence de cette lettre dans le mot est démasquée. Le jeu se termine après sept mauvais essais ou lorsque toutes les lettres ont été démasquées.

1. On commence par la fonction `mystere` suivante :

```
bool mystere(vector<char> bla, char bar) {
    for (int hop=0; hop < bla.size(); hop++)
        if ( bla[hop] == bar )
            return true;
    return false;
}
```

Deviner ce que cette fonction est sensée faire. Réécrire la fonction avec sa documentation en choisissant des noms informatifs pour elle-même et ses variables.

2. Compléter les tests pour la fonction `masqueMot` suivante, avec au moins trois appels pertinents à ASSERT. La fonction `mystere` est celle de la première question.

```
/** Masque les lettres du mot 'mot' qui ne sont pas dans 'lettres'
 * @param mot: une chaîne de caractères
 * @param lettres: un vecteur de lettres
 * @return une chaîne de caractère: le mot masqué
 **/
string masqueMot(string mot, vector<char> lettres) {
    string res = "";
    for (int i=0; i < mot.size(); i++)
        if (mystere(lettres, mot[i]))
            res += mot[i];
        else
            res += "*";
    return res;
}

void masqueMotTest() {









}
```

3. Compléter le code de la fonction `motTrouve` suivante :

```
/** Teste si toutes les lettres dans le mot ont été trouvées
 * @param mot une chaîne de caractères
 * @param lettres un tableau de lettres
 * @return vrai si toutes les lettres du mot sont dans lettres
 *         et faux sinon
 **/
bool motTrouve(string mot, vector<char> lettres) {









}
```

4. La partie principale du jeu est donnée ci-dessous :

```
1  bool partie(string mot){
2      vector<char> lettres = {};
3      int nb_essais = 0;
4      int nb_max = 7;
5      char lettre;
6
7      cout << "Bienvenu sur le Jeu du Pendu!" << endl;
8      do {
9          cout << "Mot mystère: "  << masqueMot(mot, lettres) << endl;
10         cout << "Il vous reste " << nb_max-nb_essais << " essais. "
11              << "Proposez une lettre: ";
12         cin >> lettre;
13
14         if ( mystere(lettres, lettre) )
15             cout << "Lettre déjà proposée ..." << endl;
16         else {
17             lettres.push_back(lettre);
18             if ( mystere(mot, lettre) )
19                 cout << "Bien vu!"<< endl;
20             else {
21                 nb_essais + 1;
22                 cout << "Raté, réfléchissez plus fort ..." << endl;
23             }
24             if ( not motTrouve(mot, lettres) )
25                 return true;
26         }
27     } while ( nb_essais <= nb_max );
28     return false;
29 }
30
31 int main() {
32     string mot = "chat";
33     if (partie(mot))
34         cout << "Félicitation, le mot est: " << mot << endl;
35     else
36         cout << "Désolé, le jeu est fini." << endl;
37     return 0;
38 }
```


La fonction `partie` contient des bogues à corriger l'un après l'autre. Dans chaque cas, **identifier le numéro de la ligne fautive et comment la corriger**.

(a) Dès le premier essai, le programme s'arrête en disant que le joueur a gagné.

(b) Le nombre d'essais n'est pas incrémenté à chaque tour.

(c) Le programme s'arrête au bout de huit mauvais essais et non sept.

Feuille de réponses, Questions à Choix Multiples, 23/01/2013

Écrivez votre nom et prénom ci-dessous, et codez votre numéro d'étudiant :

Nom et prénom : ....

☐0 ☐1 ☐2 ☐3 ☐4 ☐5 ☐6 ☐7 ☐8 ☐9
☐0 ☐1 ☐2 ☐3 ☐4 ☐5 ☐6 ☐7 ☐8 ☐9
☐0 ☐1 ☐2 ☐3 ☐4 ☐5 ☐6 ☐7 ☐8 ☐9
☐0 ☐1 ☐2 ☐3 ☐4 ☐5 ☐6 ☐7 ☐8 ☐9
☐0 ☐1 ☐2 ☐3 ☐4 ☐5 ☐6 ☐7 ☐8 ☐9
☐0 ☐1 ☐2 ☐3 ☐4 ☐5 ☐6 ☐7 ☐8 ☐9
☐0 ☐1 ☐2 ☐3 ☐4 ☐5 ☐6 ☐7 ☐8 ☐9
☐0 ☐1 ☐2 ☐3 ☐4 ☐5 ☐6 ☐7 ☐8 ☐9

Les réponses aux questions à choix multiple sont à donner exclusivement sur cette feuille : les réponses données sur les feuilles précédentes ne seront pas prises en compte.

Question 1 : ☐*A* ☐*B* ☐*C* ☐*D*
Question 2 : ☐*A* ☐*B* ☐*C* ☐*D*
Question 3 : ☐*A* ☐*B* ☐*C* ☐*D*
Question 4 : ☐*A* ☐*B* ☐*C* ☐*D*
Question 5 : ☐*A* ☐*B* ☐*C* ☐*D*
Question 6 : ☐*A* ☐*B* ☐*C* ☐*D*

Nom, prénom, numéro d'étudiant : .....

MPI, Info 111, Partiel du 28 octobre 2013 (deux heures)

Tous documents autorisés, sauf les annales.
Les réponses sont à donner sur le sujet. Les réponses aux questions à choix multiples doivent être données sur la dernière page. Ces questions ont une unique bonne réponse. Des points négatifs sont affectés en cas de mauvaise réponse.

Exercice 1 (Questions à choix multiples)**.**
Qu'affichent chacun des (fragments de) programmes suivants ?

Question 1

```
int x, y;
x = 5*(2/5);
y = (5*2)/5;
cout << x << ", " << y << endl;
```

A 2, 0 B 2, 2 C 0, 0 D 0, 2

Question 2

```
int g(int x, int y) {
    x = x+1;
    return y-1;
}

int main() {
    int x,y;
    x = 1;
    y = 2;
    g(x,y);
    cout << x << ", " << y << endl;
}
```

A 1, 1 B 2, 1 C *Erreur* D 1, 2

Question 3

```
int f(int x, int y) {
    x = x+1;
    return y-1;
}

int main() {
    int x,y;
    x = 1;
    y = 2;
    x = f(y,x);
    cout << x << ", " << y << endl;
}
```

A 0, 2 B 1, 1 C 2, 1 D 1, 2

Pour votre examen, imprimez de préférence les documents compilés à l'aide de auto-multiple-choice.

Question 4

```
void f(int x, int y) {
    x = x+1;
    y = y-1;
}

int main() {
    int x,y;
    x = 1;
    y = 2;
    f(y,x);
    cout << x << ", " << y << endl;
}
```

A $3, 0$ B $2, 1$ C *Erreur* D $1, 2$

Question 5

```
    int x, y;
    y = -3;
    x = -y;
    if ( y > 0 ) {
        y = 2*x;
    } else {
        y = -y;
    }
    cout << x << ", " << y << endl;
```

A $3, 3$ B $6, -3$ C $6, 3$ D $-3, -3$

Question 6

```
    int x, y;
    x = 0;
    y = 0;
    while ( x < 10 ) {
        x = x + 1;
        y = 1;
    }
    cout << x << ", " << y << endl;
```

A $9, 9$ B $0, 1$ C $10, 1$ D $9, 1$

Exercice 2 (Fonctions)**.**

1. Implanter une fonction itérative `puissanceIteratif(n, k)` qui reçoit deux entiers positifs en paramètres et renvoie n^k.

2. Implanter une fonction récursive `puissanceRecursif(n, k)` ayant le même comportement.

Exercice 3 (Jeu du pendu)**.**
On souhaite implanter le jeu du pendu, selon le principe suivant : un mot est choisi par le programme au début du jeu. Au départ, il est affiché avec toutes les lettres masquées par des étoiles. À chaque tour, le joueur choisit une lettre entre 'a' et 'z'. Chaque occurrence de cette lettre dans le mot est démasquée. Le jeu se termine après sept mauvais essais ou lorsque toutes les lettres ont été démasquées.

1. On commence par la fonction `mystere` suivante :

```
bool mystere(vector<char> bla, char bar) {
    for (int hop=0; hop < bla.size(); hop++)
        if ( bla[hop] == bar )
            return true;
    return false;
}
```

Deviner ce que cette fonction est sensée faire. Réécrire la fonction avec sa documentation en choisissant des noms informatifs pour elle-même et ses variables.

2. Compléter les tests pour la fonction `masqueMot` suivante, avec au moins trois appels pertinents à ASSERT. La fonction `mystere` est celle de la première question.

```
/** Masque les lettres du mot 'mot' qui ne sont pas dans 'lettres'
 * @param mot: une chaîne de caractères
 * @param lettres: un vecteur de lettres
 * @return une chaîne de caractère: le mot masqué
 **/
string masqueMot(string mot, vector<char> lettres) {
    string res = "";
    for (int i=0; i < mot.size(); i++)
        if (mystere(lettres, mot[i]))
            res += mot[i];
        else
            res += "*";
    return res;
}

void masqueMotTest() {









}
```

3. Compléter le code de la fonction `motTrouve` suivante :

```
/** Teste si toutes les lettres dans le mot ont été trouvées
 * @param mot une chaîne de caractères
 * @param lettres un tableau de lettres
 * @return vrai si toutes les lettres du mot sont dans lettres
 *         et faux sinon
 **/
bool motTrouve(string mot, vector<char> lettres) {









}
```

4. La partie principale du jeu est donnée ci-dessous :

```
1  bool partie(string mot){
2      vector<char> lettres = {};
3      int nb_essais = 0;
4      int nb_max = 7;
5      char lettre;
6
7      cout << "Bienvenu sur le Jeu du Pendu!" << endl;
8      do {
9          cout << "Mot mystère: "  << masqueMot(mot, lettres) << endl;
10         cout << "Il vous reste " << nb_max-nb_essais << " essais. "
11              << "Proposez une lettre: ";
12         cin >> lettre;
13
14         if ( mystere(lettres, lettre) )
15             cout << "Lettre déjà proposée ..." << endl;
16         else {
17             lettres.push_back(lettre);
18             if ( mystere(mot, lettre) )
19                 cout << "Bien vu!"<< endl;
20             else {
21                 nb_essais + 1;
22                 cout << "Raté, réfléchissez plus fort ..." << endl;
23             }
24             if ( not motTrouve(mot, lettres) )
25                 return true;
26         }
27     } while ( nb_essais <= nb_max );
28     return false;
29 }
30
31 int main() {
32     string mot = "chat";
33     if (partie(mot))
34         cout << "Félicitation, le mot est: " << mot << endl;
35     else
36         cout << "Désolé, le jeu est fini." << endl;
37     return 0;
38 }
```


La fonction `partie` contient des bogues à corriger l'un après l'autre. Dans chaque cas, **identifier le numéro de la ligne fautive et comment la corriger**.

(a) Dès le premier essai, le programme s'arrête en disant que le joueur a gagné.

(b) Le nombre d'essais n'est pas incrémenté à chaque tour.

(c) Le programme s'arrête au bout de huit mauvais essais et non sept.

Feuille de réponses, Questions à Choix Multiples, 23/01/2013

Écrivez votre nom et prénom ci-dessous, et codez votre numéro d'étudiant :

Nom et prénom : . . . .

☐**0** ☐**1** ☐**2** ☐**3** ☐**4** ☐**5** ☐**6** ☐**7** ☐**8** ☐**9**
☐**0** ☐**1** ☐**2** ☐**3** ☐**4** ☐**5** ☐**6** ☐**7** ☐**8** ☐**9**
☐**0** ☐**1** ☐**2** ☐**3** ☐**4** ☐**5** ☐**6** ☐**7** ☐**8** ☐**9**
☐**0** ☐**1** ☐**2** ☐**3** ☐**4** ☐**5** ☐**6** ☐**7** ☐**8** ☐**9**
☐**0** ☐**1** ☐**2** ☐**3** ☐**4** ☐**5** ☐**6** ☐**7** ☐**8** ☐**9**
☐**0** ☐**1** ☐**2** ☐**3** ☐**4** ☐**5** ☐**6** ☐**7** ☐**8** ☐**9**
☐**0** ☐**1** ☐**2** ☐**3** ☐**4** ☐**5** ☐**6** ☐**7** ☐**8** ☐**9**
☐**0** ☐**1** ☐**2** ☐**3** ☐**4** ☐**5** ☐**6** ☐**7** ☐**8** ☐**9**

Les réponses aux questions à choix multiple sont à donner exclusivement sur cette feuille : les réponses données sur les feuilles précédentes ne seront pas prises en compte.

Question 1 : ☐*A* ☐*B* ☐*C* ☐*D*
Question 2 : ☐*A* ☐*B* ☐*C* ☐*D*
Question 3 : ☐*A* ☐*B* ☐*C* ☐*D*
Question 4 : ☐*A* ☐*B* ☐*C* ☐*D*
Question 5 : ☐*A* ☐*B* ☐*C* ☐*D*
Question 6 : ☐*A* ☐*B* ☐*C* ☐*D*