

1 TP3

1.1 Exercice 1

Écrire un programme qui permet à l'utilisateur de saisir la largeur et la longueur d'un champ et qui affiche le périmètre et la surface de celui-ci.

- Les variables largeur et longueur seront de type réels double précision
- Le calcul du périmètre se fera dans une fonction appelée `perimetre`, de même que le calcul de la surface se fera dans une fonction appelée `surface`.

```
1  //-----
2  //TP : Perimetre
3  //-----
4  #include "/usr/local/public/BIBLIOC++/entreeSortie.h"
5  #include "/usr/local/public/BIBLIOC++/chaine.h"
6
7  int perimetre (int a, int b);
8  int surface (int a, int b);
9
10 int main ()
11 {
12     int l;          // largeur
13     int L;          // Longueur
14
15     ecrire (uneChaine ("Donnez la valeur de la largeur : "));
16     lire (l);
17     ecrire (uneChaine ("Donnez la valeur de la longueur : "));
18     lire (L);
19     ecrire (uneChaine ("Le perimetre est de "));
20     ecrireNL (perimetre (l,L));
21     ecrire (uneChaine ("La surface est de "));
22     ecrireNL (surface (l,L));
23     return 0 ;
24 }
25
26 int perimetre (int a,int b)
27 {
28     return (a+b)*2;
29 }
30
31 int surface (int a, int b)
32 {
33     return (a*b);
34 }
```

Listing 1 – Périmètre et surface

1.2 Exercice 2

- Écrire un programme qui demande à l'utilisateur de renseigner 5 entiers et qui affiche leur moyenne.
- Le programme ne devra utiliser que 2 variables.

```
1  //-----
2  //TP : moyenne
3  //-----
4  #include "/usr/local/public/BIBLIOC++/entreeSortie.h"
5  #include "/usr/local/public/BIBLIOC++/chaine.h"
6
7  int main ()
8  {
9      int entier;
10     int moyenne = 0;
11
12     ecrire (uneChaine("Entrer une note : "));
13     lire (entier);
14     moyenne = (moyenne + entier);
15
16     ecrire (uneChaine("Entrer une note : "));
17     lire (entier);
18     moyenne = (moyenne + entier);
19
20     ecrire (uneChaine("Entrer une note : "));
21     lire (entier);
22     moyenne = (moyenne + entier);
23
24     ecrire (uneChaine("Entrer une note : "));
25     lire (entier);
26     moyenne = (moyenne + entier);
27
28     ecrire (uneChaine("Entrer une note : "));
29     lire (entier);
30     moyenne = ((moyenne + entier)/5);
31
32     ecrire (uneChaine("La moyenne de l' lve est de "));
33     ecrireNL (moyenne);
34 }
```

Listing 2 – Moyenne

1.3 Exercice 3

- Écrire un programme qui demande à l'utilisateur de renseigner 2 entiers et qui affiche leur somme.
- Le calcul de la somme se fera dans une procédure appelée calculerSomme ayant les 2 entiers lus en entrée et l'entier résultat en sortie.

```
1  //-----
2  //TP : somme
3  //-----
4  #include "/usr/local/public/BIBLIOC++/entreeSortie.h"
5  #include "/usr/local/public/BIBLIOC++/chaine.h"
6
7  void calculerSomme (int a, int b, int &resultat)
8  {
9      resultat = a + b;
10 }
11
```

```

12  int main ()
13  {
14      int entier1;
15      int entier2;
16      int somme = 0;
17      ecrire (uneChaine("Ecrire 1 : "));
18      lire (entier1);
19      ecrire (uneChaine ("Ecrire 2 : "));
20      lire (entier2);
21      calculerSomme (entier1, entier2, somme);
22      ecrire (somme);
23
24      return 0;
25  }

```

Listing 3 – Somme

2 TP4

2.1 Exercice 1

Ecrire un programme qui demande à un utilisateur de saisir un caractère et qui affiche si c'est un caractère minuscule, majuscule ou un chiffre

- La fonction `estMinuscule` prend un caractère en entrée et renvoie un booléen. Même chose pour les fonctions `estMajuscule` et `estChiffre`
- Les trois fonctions précédentes s'écrivent en une seule instruction !

```

1  //-----
2  //TP : MAJ
3  //-----
4  #include "/usr/local/public/BIBLIOC++/entreeSortie.h"
5  #include "/usr/local/public/BIBLIOC++/chaine.h"
6
7  bool estMinuscule (char a)
8  {
9      return (a >='a' && a <= 'z');
10 }
11
12 bool estMajuscule (char a)
13 {
14     return (a >='A' && a <= 'Z');
15 }
16
17 bool estUnChiffre (int a)
18 {
19     return (a >= 0 && a <= 9);
20 }
21
22 int main ()
23 {
24     char a;
25
26     ecrire (uneChaine("Saisir un caract re : "));
27     lire (a);
28
29     if (estMinuscule (a))
30     {
31         ecrire (uneChaine("le caractere entr est une minuscule \n"));

```

```

32     } else if (estMajuscule (a)) {
33         ecrire (uneChaine("le caractere entr  est une majsucule\n"));
34     } else {
35         ecrire (uneChaine ("est un chiffre\n"));
36     }
37
38     return 0;
39 }

```

Listing 4 – Détermine si c'est une majuscule

2.2 Exercice 2

Ecrire un programme qui demande à un utilisateur de saisir un caractère et qui affiche son code en ASCII, puis qui demande à un utilisateur de saisir un code ASCII sous forme entière et affiche le caractère correspondant

- Les codes des conversions entre type caractère et type entier ne doivent pas dépasser une ligne de C++.

```

1  //-----
2  //TP : ASCII
3  //-----
4  #include "/usr/local/public/BIBLIOC++/entreeSortie.h"
5  #include "/usr/local/public/BIBLIOC++/chaine.h"
6
7  #include <iostream>
8
9  int main() {
10     int x;
11     char y;
12
13     ecrire (uneChaine("Ecrire une lettre : "));
14     lire (y);
15     printf ("%d", y);
16
17     ecrire (uneChaine("Ecrire un chiffre : "));
18     lire (x);
19     putchar (x);
20     ecrire (uneChaine("\n"));
21     return 0 ;
22 }

```

Listing 5 – Détermine le code ASCII

2.3 Exercice 3

Ecrire une fonction `changerMinusculesMajuscules` qui a un paramètre de type chaîne de caractères (en mise à jour) et qui transforme toutes les minuscules de la chaîne en majuscules et toutes les majuscules en minuscules. Tester en demandant à un utilisateur de saisir une chaîne et afficher alors la chaîne modifiée.

- Vous utiliserez le type `Chaine` de `chaine.h`
- Les traitements sur les caractères de la chaîne seront encapsulés dans les fonctions `minusculeEnMajuscule` et `majusculeEnMinuscule`
- Les chaînes de caractères sont indicées de 0 à `longueur(chaine) - 1`
- Vous utiliserez une boucle `for` pour traiter tous les caractères de la chaîne.

```

1  //-----
2  //TP : minToMaj
3  //-----
4  #include "/usr/local/public/BIBLIOC++/entreeSortie.h"
5  #include "/usr/local/public/BIBLIOC++/chaine.h"
6

```

```

7  bool estMinuscule (char a)
8  {
9      return (a >='a' && a<= 'z');
10 }
11
12 bool estMajuscule (char a)
13 {
14     return (a >='A' && a<= 'Z');
15 }
16
17 int main ()
18 {
19     Chaine x;
20
21     ecrire (uneChaine("entrer une chaine de caractere : "));
22     lire (x);
23
24     for (int i = 0 ; i < longueur(x) ; i++)
25     {
26         if (estMinuscule (x.chaine[i]))
27         {
28             x.chaine[i] = 'A' + (x.chaine[i] - 'a') ;
29         }
30         else if (estMajuscule (x.chaine[i]))
31         {
32             x.chaine[i] = 'a' + (x.chaine[i] - 'A') ;
33         }
34     }
35
36     ecrireNL (x) ;
37
38     return 0 ;
39 }
40

```

Listing 6 – fonction changerMinusculeMajuscules

2.4 Exercice 4

Ecrire un programme qui demande à l'utilisateur de taper une chaîne de caractères et qui affiche la ou les lettres les plus fréquentes. Ecrire un programme qui demande à l'utilisateur de taper une chaîne de caractères et qui affiche la ou les lettres les plus fréquentes.

```

1  //-----
2  //TP : frequente
3  //-----
4  #include "/usr/local/public/BIBLIOC++/entreeSortie.h"
5  #include "/usr/local/public/BIBLIOC++/chaine.h"
6
7  int main ()
8  {
9      Chaine x;
10     char caractereActuel ;
11     int iMax = 0 ;
12     int iCompteur = 0 ;
13     int iRang = 0 ;
14
15

```

```

16   ecrire (uneChaine("entrer une chaine de caractere : "));
17   lire (x);
18
19   for (int i = 0 ; i < longueur (x) ; i++)
20   {
21       caractereActuel = ieme(x, i) ;
22       iCompteur = 0 ;
23
24       for (int j = i ; j < longueur(x) ; j++) {
25           if (ieme(x, j) == caractereActuel) {
26               iCompteur ++ ;
27           }
28       }
29
30       if (iCompteur > iMax) {
31           iMax = iCompteur ;
32           iRang = i ;
33       }
34   }
35
36   ecrire(uneChaine("Le caractere le plus r current est : ")) ;
37   ecrireNL(ieme(x, iRang)) ;
38   ecrire(uneChaine("Il apparait ")) ;
39   ecrire(iMax) ;
40   ecrireNL(uneChaine(" fois.")) ;
41
42
43 }

```

Listing 7 – Lettre la plus fréquente

3 TP5

3.1 Exercice 1

Ecrire un programme qui demande à un utilisateur de saisir 4 entiers A, B, C et D, représentant respectivement les intervalles d'entiers [A,B] et [C,D] puis qui affiche quelle est l'intersection des intervalles [AB] et [CD].

- Déclarez et utilisez une structure Intervalle comprenant deux entiers inf et sup.
- Ecrire une procédure chargée de la saisie d'un intervalle.
- Ecrire une procédure chargée de l'affichage d'un intervalle.
- Ecrire une fonction intersection prenant deux intervalles en entrée et renvoyant comme résultat l'intervalle représentant l'intersection.
- La constante INTERVALLE_VIDE égale à [0,0] représentera l'intersection vide.

```

1  //-----
2  //TP : Intervalles
3  //-----
4  #include "/usr/local/public/BIBLIOC++/entreeSortie.h"
5  #include "/usr/local/public/BIBLIOC++/chaine.h"
6
7  // Enregistrement d'un intervalle
8
9  struct Intervalle
10 {
11     int inf;
12     int sup;
13 };
14

```

```
15
16 // Proc dure charg e de la saisie d'un intervalle
17
18 void saisirIntervalle (Intervalle &i)
19 {
20     int iBuffer ;
21
22     ecrire (uneChaine("Borne inf ?"));
23     lire(iBuffer) ;
24     i.inf = iBuffer;
25     ecrire (uneChaine("Borne sup ?"));
26     lire(iBuffer) ;
27     i.sup = iBuffer;
28 }
29
30
31 // Proc dure charg e d'afficher un intervalle
32
33 void afficherIntervalle (const Intervalle intervalle1)
34 {
35     ecrire (uneChaine("["));
36     ecrire (intervalle1.inf);
37     ecrire (uneChaine(","));
38     ecrire (intervalle1.sup);
39     ecrire (uneChaine("]"));
40 }
41
42 // Fonction intersection renvoyant l'intersection de deux intervalles
43
44 Intervalle intersection (Intervalle intervalle1, Intervalle intervalle2)
45 {
46     Intervalle inter = {0,0} ;
47
48     if (intervalle1.sup < intervalle2.inf) {
49         ecrire(uneChaine("Ensemble vide")) ;
50     } else {
51         if(intervalle1.inf > intervalle2.sup) {
52             ecrire(uneChaine("Ensemble vide"));
53         } else {
54             inter.inf = intervalle2.inf ;
55             inter.sup = intervalle1.sup ;
56         }
57     }
58
59
60     return inter ;
61 }
62
63
64 // Programme
65
66 int main (void)
67 {
68     Intervalle i1, i2 ;
69
70     saisirIntervalle(i1) ;
71     saisirIntervalle(i2) ;
72
```

```

73     afficherIntervalle(i1) ;
74     afficherIntervalle(i2) ;
75
76     afficherIntervalle(intersection(i1,i2)) ;
77
78
79     return 0;
80 }

```

Listing 8 – Intervalles

3.2 Exercice 2

Sachant que : l'eau gèle à $0^{\circ}C$, le fuel gèle à $-5^{\circ}C$, le super gèle à $-23^{\circ}C$, l'ordinaire gèle à $-13^{\circ}C$ et l'eau salée gèle à $-3^{\circ}C$. Ecrire un programme qui demande à un utilisateur de saisir une température et qui affiche la liste des liquides gelés à cette température. Ecrire un programme qui gère un menu. L'affichage de ce menu comporte 4 choix d'actions possible. La dernière de ces actions est la sortie du programme. A chaque action sélectionnée par un utilisateur, il est affiché le nom de l'action considérée, par exemple action1. Puis l'utilisateur tape sur la touche entrée et le menu est de nouveau affiché si le choix n'est pas la sortie du programme.

- La gestion des actions en fonction des choix effectués par l'utilisateur se programmera à l'aide d'une instruction `switch(choix)`.
- L'affichage du menu se répètera tant que l'utilisateur n'a pas décidé de sortir du programme.
- La gestion des erreurs de saisie sera prise en compte. On supposera que l'utilisateur saisit toujours un entier.

```

1  //-----
2  //TP : gele
3  //-----
4  #include "/usr/local/public/BIBLIOC++/entreeSortie.h"
5  #include "/usr/local/public/BIBLIOC++/chaine.h"
6
7  int iTemp; // Temp rature
8
9  int main ()
10 {
11     ecrire (uneChaine("Voici la liste des liquides : \n- eau \n- fuel\n- super
        \n- ordinaire \n- eau sal e\n\n"));
12     ecrire (uneChaine("Saisir une temp rature : "));
13     lire (iTemp);
14     switch (iTemp)
15     {
16         case (-23) : ecrireNL(uneChaine("Le super g le"));
17         case (-13) : ecrireNL(uneChaine("L'ordinaire g le"));
18         case (-5) : ecrireNL(uneChaine("Le fuel g le"));
19         case (0) : ecrireNL(uneChaine("L'eau g lee"));
20         break;
21         default : ecrireNL(uneChaine("Aucun des liquides ne g lent      cette
        temp rature"));
22     }
23 }

```

Listing 9 – Gèle des liquides

3.3 Exercice 3

Ecrire un programme qui gère un menu. L'affichage de ce menu comporte 4 choix d'actions possible. La dernière de ces actions est la sortie du programme. A chaque action sélectionnée par un utilisateur, il est affiché le nom de l'action considérée, par exemple action1. Puis l'utilisateur tape sur la touche entrée et le menu est de nouveau affiché si le choix n'est pas la sortie du programme.

- La gestion des actions en fonction des choix effectués par l'utilisateur se programmera à l'aide d'une instruction switch(choix).
- L'affichage du menu se répètera tant que l'utilisateur n'a pas décidé de sortir du programme.
- La gestion des erreurs de saisie sera prise en compte. On supposera que l'utilisateur saisit toujours un entier.

```

1  //-----
2  //TP : Menu
3  //-----
4  #include "/usr/local/public/BIBLIOC++/entreeSortie.h"
5  #include "/usr/local/public/BIBLIOC++/chaine.h"
6
7  int iMenu ;      //Choix du menu
8
9  int main ()
10 {
11     ecrire (uneChaine("Voici la liste du menu : \n- Choix 1 : Se suicider\n-
        Choix 2 : Sauter par la fenetre\n- Choix 3 : Se tirer une balle\n-
        Choix 4 : rester en vie"));
12     ecrire (uneChaine("Saisir un choix en entrant le numero du choix : "));
13     lire (iMenu);
14     ecrire(uneChaine("\n"));
15     while (iMenu < 1 || iMenu > 4)
16     {
17         ecrireNL(uneChaine("Erreur - Saisir un choix en entrant le numero du
            choix compris entre 1 et 4 : "));
18         lire (iMenu);
19     }
20
21     switch (iMenu)
22     {
23         case (1):
24             ecrireNL(uneChaine("Vous avez choisi de vous suicider"));
25             break;
26         case (2):
27             ecrireNL(uneChaine("Vous avez choisi de sauter par la fenetre"));
28             break;
29         case (3):
30             ecrireNL(uneChaine("Vous avez choisi de vous tirer une balle"));
31             break;
32         case (4):
33             ecrireNL(uneChaine("Vous avez choisi de rester en vie"));
34             break;
35     }
36 }

```

Listing 10 – Gestion d'un menu

4 TP6

Ecrire un programme qui demande à un utilisateur de saisir un entier N et qui affiche la figure suivante. Vous utiliserez uniquement des boucles for () ... imbriquées. N=1

N=2

*

N=3

**

*

```

1  //-----
2  //TP : figures
3  //-----
4  #include "/usr/local/public/BIBLIOC++/entreeSortie.h"
5  #include "/usr/local/public/BIBLIOC++/chaine.h"
6
7  int main ()
8  {
9      int iNombre;
10     int cpt=0;
11     int cpt2 = 0;
12     int u = iNombre;
13
14     ecrire(uneChaine("Entrer le nombre d' toile sur la premiere ligne : "));
15     lire (iNombre);
16     ecrire(uneChaine("\n"));
17
18     for (int i = 0 ; i < iNombre ; i++) {
19         for (int j = 0 ; j < i+1 ; j++) {
20             ecrire(uneChaine("*")) ;
21         }
22         ecrireNL() ;
23     }
24 }

```

5 TP7

5.1 Exercice 1

Déclarez un tableau de 20 entiers et écrivez deux procédures permettant de renseigner et afficher les éléments de ce tableau.

- La procédure permettant de saisir une suite de 20 valeurs entières et de les affecter à chaque poste de tableau s'appellera saisirTableau.
- La procédure permettant d'afficher les 20 valeurs entières contenues dans le tableau s'appellera afficherTableau.
- Testez vos programmes dans un programme principal

```

1  -----
2  TP : Tableau_ex1
3  -----
4  #include "/usr/local/public/BIBLIOC++/entreeSortie.h"
5  #include "/usr/local/public/BIBLIOC++/chaine.h"
6
7  /***** ENREGISTREMENT DE TYPE DATE *****/
8  struct Date
9  {
10     int jour;
11     int mois;
12     int annee;
13 };
14
15 Chaine mois[12] = { "Janvier", "Fevrier", "Mars", "Avril", "Mai", "Juin", "
    Juillet" , "Aout" , "Septembre" , "Octobre" , "Novembre" , "Decembre" };

```

```

16
17 /***** PROCEDURE DE SAISIE D'UNE DATE *****/
18 void saisirDate (Date &date1)
19 {
20     ecrire(uneChaine("Veuillez saisir un jour[1-31] "));
21     lire (date1.jour);
22     ecrire(uneChaine("Veuillez saisir un mois[1-12] "));
23     lire (date1.mois);
24     ecrire(uneChaine("Veuillez saisir une ann e[AAAA] "));
25     lire (date1.annee);
26 }
27
28
29 /***** AFFICHAGE D'UNE DATE *****/
30 void afficherDate (Date &date1)
31 {
32     ecrire(date1.jour);
33     ecrire(uneChaine(" ")) ;
34     ecrire(mois[date1.mois-1]);
35     ecrire(uneChaine(" ")) ;
36     ecrireNL(date1.annee);
37 }
38
39
40 /***** PROGRAMME *****/
41 int main (void)
42 {
43     Date date;
44
45     saisirDate (date);
46     afficherDate (date);
47
48     return 0 ;
49 }

```

5.2 Exercice 3 & 4

fonction trierParSelection et trierABulles.

```

1  #include <string.h>
2  #include <iostream>
3  using namespace std;
4
5  void trierABulles(int *tab, int tailleTab);
6  void trierParSelection(int *tab, int tailleTab);
7
8  int main (void)
9  {
10     int tab[11] = {0, 1, 6548, 3, 2, 21, 54, 23, 4, 235, 24}; //initialisation
11     du tableau      trier
12     trierParSelection(tab, 10); //tri par s l ction
13     // trierABulles(tab, 10)- //tri bulle;
14
15     /* affichage du tableau tri */
16     for (int i = 0; i <= 10; i++)
17     {
18         cout << tab[i] << endl;
19     }

```

```

19 }
20 void trierParSelection(int *tab, int tailleTab)
21 {
22     int indicePetitElement = tab[0], echange = 0;
23     for(int j = 0; j < tailleTab; j++)
24     {
25         indicePetitElement = j;
26         for(int i = j; i <= tailleTab; i++)
27         {
28             /* on cherche le plus petit lment */
29             if(tab[i] < tab[indicePetitElement])
30             {
31                 indicePetitElement = i;
32             }
33         }
34
35         echange = tab[indicePetitElement] ;
36         tab[indicePetitElement] = tab[j];
37         tab[j] = echange;
38     }
39 }
40 }
41
42 void trierABulles(int *tab, int tailleTab)
43 {
44     int echange = 0;
45     bool tri = 0;
46
47     while(!tri)
48     {
49         tri = 1;
50         for (int i = 0; i <= tailleTab; i++)
51         {
52             if((tab[i]) > (tab[i+1]))
53             {
54                 echange = tab[i];
55                 tab[i] = tab[i+1];
56                 tab[i+1] = echange;
57                 tri=0;
58             }
59         }
60     }
61 }
62 }

```

6 TP8

définir une date sous la forme 6 janvier 2010 par une structure à trois champs. Ecrire des sous-programmes de saisie et d'affichage d'une date. Vous pourriez gérer la validité d'une date saisie ! Réutilisez cette date pour définir une personne comme une structure composée d'un nom, d'un prénom et d'une date de naissance. Ecrire des sous-programmes de saisie et d'affichage d'une personne.

```

1  #include <string.h>
2  #include <iostream>
3  using namespace std;
4
5  struct Date

```

```
6  {
7    int jour;
8    int mois;
9    int annee;
10 };
11
12 int main (void)
13 {
14     Date aujourd'hui = {1, 1, 2010};
15
16     cout << "Nous sommes le " << aujourd'hui.jour << "/" << aujourd'hui.mois <<
17         "/" << aujourd'hui.annee << endl;
18
19     cout << "Ou plut t le " << aujourd'hui.jour ;
20     switch(aujourd'hui.mois)
21     {
22         case 1:
23             cout << "Janvier";
24             break;
25         case 2:
26             cout << "F vrier";
27             break;
28         case 3:
29             cout << "Mars";
30             break;
31         case 4:
32             cout << "Avril";
33             break;
34         case 5:
35             cout << "Mai";
36             break;
37         case 6:
38             cout << "Juin";
39             break;
40         case 7:
41             cout << "Juillet";
42             break;
43         case 8:
44             cout << "Aout";
45             break;
46         case 9:
47             cout << "Septembre";
48             break;
49         case 10:
50             cout << "Octobre";
51             break;
52         case 11:
53             cout << "Novembre";
54             break;
55         case 12:
56             cout << "D cembre";
57             break;
58     }
59     cout << aujourd'hui.annee << endl;
60 }
```