

**QUESTION 1 (3 Points)**

**1.1** (0,5 point) Quelle est la valeur en décimal (base 10) du nombre entier 10000000, représenté en complément à 2.

**1.2** (1 point) Donner le résultat de l'opération binaire de l'addition de deux nombres entiers représentés en compléments à 2 :  $10010000 + 11001010$ . S'agit-il d'un débordement, si oui pourquoi ?

**1.3** (1,5 points) Donner la représentation du nombre réel 0,2 selon la norme IEEE754 simple précision en précisant clairement le signe, la mantisse et l'exposant.

---

**QUESTION 2 (7,5 Points)**

Un programme réalise la gestion du trafic aérien d'un aéroport. Le programme utilise la structure de données suivante pour la mémorisation et la gestion de l'information:

```
enum Depart_Arrivee {DEPART, ARRIVEE};

struct Heure
{
    int heures;
    int minutes;
};

struct Vol
{
    char id[9];
    Depart_Arrivee depOuArr;
    Heure heurePrevue;
    Heure heureRevisee;
    char etatVol;
};
```

Nous vous conseillons de détacher cette page du questionnaire pour avoir en tout temps ces déclarations qui vous seront fortement utiles pour répondre aux sous-questions suivantes.

Ce programme utilise plusieurs fonctions. Compléter les trois fonctions suivantes en respectant précisément la description.

**Inscrire dans votre cahier d'examen le numéro de la question (2.1, 2.2, etc.) suivi de votre réponse.**

**(SUITE À LA PAGE SUIVANTE)**

**Création du fichier binaire**

La fonction `creerFichierBinaire()` lit un fichier texte nommé `Vols.txt` et transcrit les informations lues dans un fichier binaire nommé `Vols.bin` sous la forme d'un enregistrement de type `Vol` pour chaque vol.

Le fichier texte `Vols.txt` est constitué de plusieurs lignes, chacune contenant l'indicatif d'un vol (ex : AF346, 5 à 8 caractères), suivi d'une lettre indiquant s'il s'agit d'un départ (D) ou d'une arrivée (A), l'heure de départ ou d'arrivée prévue sous forme de deux entiers consécutifs (ex : 9 30 pour 09h30 et 15 57 pour 15h57), l'heure de départ ou d'arrivée révisée, et l'état du vol qui peut avoir l'un des trois caractères suivants (A signifie à temps, N signifie vol annulé, R signifie vol retardé).

Exemple d'une ligne du fichier texte :

```
AMEX311 D 9 30 9 30 A
AIRC911 A 15 30 16 00 R
```

```
void creerFichierBinaire( )
{
    ifstream fichierTexte;
    fstream fichierBinaire;

    char departOuArrivee; //Permet de lire D ou A dans le fichier

    Vol unVol;

    fichierTexte.open("Vols.txt");

    //2.1 (0,25 point) Ouverture du fichier binaire Vol.bin en
    //écriture. Vous n'avez pas à vérifier l'ouverture du fichier.

    INSTRUCTION(S) À COMPLÉTER

    //2.2 (2,5 points) Lecture du fichier texte Vol.txt et
    //transcription dans le fichier binaire Vol.bin.

    INSTRUCTION(S) À COMPLÉTER

    fichierTexte.close();
    fichierBinaire.close();
}
```

**(SUITE À LA PAGE SUIVANTE)**

**Obtenir les informations d'un vol**

---

La fonction `obtenirVol()` recherche dans le fichier binaire `Vol.bin` un vol donné et le retourne après sa lecture à l'aide d'un paramètre transmis par adresse. La fonction reçoit en paramètre la position de l'enregistrement du vol dans le fichier. Précisons que le paramètre position vaut 1 pour le premier enregistrement dans le fichier, 2 pour le deuxième, et ainsi de suite. La fonction retourne la valeur `true` si la position est acceptable selon le nombre d'enregistrements dans le fichier ou `false` autrement. Le fichier binaire est déjà correctement ouvert.

```
bool obtenirVol(int position, Vol &unVol)
{
    fstream fichierBinaire;
    bool positionOK = true;

    //Le fichier binaire est déjà correctement ouvert

    //2.3 (1 point) Vérifier que position n'est pas plus grand que
    //le nombre d'enregistrements dans le fichier, sinon attribuer
    //à la variable positionOK la valeur false

    INSTRUCTION(S) À COMPLÉTER

    //2.4 (1 point) Lire l'enregistrement situé à la position
    //précisée en paramètre.

    INSTRUCTION(S) À COMPLÉTER

    return positionOK;
}
```

(SUITE À LA PAGE SUIVANTE)

**Réviser l'état d'un vol**

---

La fonction `reviserUnVol()` effectue une mise à jour du fichier binaire `Vol.bin` en modifiant l'état d'un vol. La fonction reçoit en paramètre l'identificateur du vol à modifier ainsi qu'un caractère correspondant au nouvel état du vol. Une fois l'enregistrement du vol trouvé, la fonction modifie l'enregistrement en inscrivant la nouvelle information. L'enregistrement mis à jour est réécrit au même emplacement dans le fichier. La fonction retourne `true` si l'identificateur de vol recherché est bien dans le fichier, `false` autrement.

```
bool reviserUnVol (const char idVolCherche[9], char nouvelEtat)
{
    fstream fichierBinaire;
    int nbVols; // nombre de vols dans le fichier
    Vol unVol;
    bool estModifie = false;

    //2.5 (0,50 point) Ouvrir le fichier binaire en lecture et en
    //écriture. Vous n'avez pas à vérifier l'ouverture du fichier.
```

**INSTRUCTION(S) À COMPLÉTER**

```
    nbVols = // considérer que cette variable contient le nombre de
              // vols dans le fichier - aucune instruction à ajouter.

    //2.6 (2,25 points) Rechercher le vol ayant idVolCherche comme
    //identificateur de vol. S'il est présent, remplacer l'état par
    //nouvelEtat et réécrire l'enregistrement modifié dans le fichier
    //binaire. S'il est absent, affecter la valeur false à la
    //variable estModifie. Vous pouvez utiliser la fonction strcmp()
    //pour comparer deux tableaux de caractères. strcmp() retourne 0
    //si les chaînes sont identiques.

    INSTRUCTION(S) À COMPLÉTER

    return estModifie;
}
```

**QUESTION 3 (4 Points)**

Donner l’affichage obtenu après l’exécution du programme suivant.

<pre>#include &lt;iostream&gt; #include &lt;string&gt; using namespace std;  void f1(int tab[]) {     int i;     string mot="cabacabe";      for (i=0; i&lt;mot.size();i++)         tab[mot[i]-'a']++; }  int f2(char a,int x) {     int i,j;     string mot="enfin fini";      j=0;     x=0;     for (i=0; i&lt;mot.size();i++)         if (mot[i]==a)             j++;         else             x++;     return j; }  void f3(char a, char&amp; b, char&amp; c) {     b = a;     a = c;     c = b; }  void f4(string mot1, string mot2,         string* tab) {     int i=0;      tab[0]=mot1;     tab[1]=mot2;     if (mot1&lt;mot2)         tab[i]=mot2;     i++;     tab[i]=mot1; }</pre>	<pre>int main () {     int tableau[26];     int i, j;     for (i=0;i&lt;26;i++)         tableau[i]=0;     f1(tableau);     cout &lt;&lt; "Reponse f1 : " &lt;&lt; endl;     for (i=0;i&lt;26;i++)         if (tableau[i]!=0)             cout &lt;&lt; i &lt;&lt; ' '                 &lt;&lt; tableau[i] &lt;&lt; endl;      char lettre = 'i';     j=0;     i = f2(lettre,j);     cout &lt;&lt; "Reponse f2 : " &lt;&lt; endl;     cout &lt;&lt; i &lt;&lt; ' ' &lt;&lt; lettre &lt;&lt; ' '         &lt;&lt; j &lt;&lt; endl;      char a='a', b='b', c='c';     f3(a,b,c);     cout &lt;&lt; "Reponse f3 : " &lt;&lt; endl;     cout &lt;&lt; a &lt;&lt; ' ' &lt;&lt; b &lt;&lt; ' '         &lt;&lt; c &lt;&lt; endl;      string m1="bon", m2="conge";     string* tabString = new string[3];     for (i=0;i&lt;3;i++)         tabString[i]="bye bye";     f4(m1,m2,tabString);     cout &lt;&lt; "Reponse f4 : " &lt;&lt; endl;     cout &lt;&lt; m1 &lt;&lt; ' ' &lt;&lt; m2 &lt;&lt; ' '         &lt;&lt; endl;     for (i=0;i&lt;3;i++)         cout &lt;&lt; tabString[i] &lt;&lt; ' ';      delete []tabString;     return 0; }</pre>
---	---

**QUESTION 4 (5,5 Points)**

Compléter la fonction `lireFichier()` qui reçoit en paramètre le nom d'un fichier texte à lire, lit le fichier et mémorise l'information sur chaque artiste à l'aide d'un tableau dynamique de type `Artiste` défini ci-dessous. La fonction retourne le nombre d'artistes et le tableau dynamique via deux autres paramètres. Finalement, la fonction retourne la valeur `false` en cas d'erreur ou `true` s'il n'y a pas d'erreur.

```
struct Album
{
    string nomAlbum;
    int annee ;
};

struct Artiste
{
    string nom;
    string langue;
    int nombreAlbums;
    Album *listeAlbums;

};
```

L'entête de la fonction `lireFichier()` est

```
bool lireFichier(string nomFichier, int& nbArtistes, Artiste*& artistes)
```

Le fichier texte à lire contient sur chaque ligne le nom de l'artiste, la langue soit française ou internationale, le nombre d'albums à considérer, la liste des titres de chaque album avec leur année de sortie. Dans ce fichier, tous les espaces dans le nom des artistes et le titre des albums sont remplacés par des '\_'. Voici un exemple de fichier :

```
Freddy_Mercury internationale 2 made_in_heaven 1995 the_miracle 1989
Roch_voisine française 2 americana 2010 sauf_si_l'amour 2005
Coldplay internationale 1 viva_la_vida 2008
Ariane_mofatt française 2 tous_les_sens 2008 le_coeur_dans_la_tête 2005
```

**Inscrire dans votre cahier d'examen le numéro de la question (4.1, 4.2, etc.) suivi de votre réponse.**

**(SUITE À LA PAGE SUIVANTE)**

```
bool lireFichier(string nomFichier, int& nbArtistes, Artiste*& artistes)
{
    ifstream fic;          // fichier des artistes
    string nom,            // nom de l'artiste
           ligne,          // permet de lire une ligne complète
           langue;         // langue : français ou international
    int nbAlbums;          // nombre d'albums d'un artiste
    bool estLu = false;    // le fichier est lu ou non

    fic.open(nomFichier.c_str());

    if (!fic.fail())
    {
        nbArtistes = 0;

        //4.1 (1,25 points) Déterminer le nombre d'artistes
        //présents dans le fichier

        INSTRUCTION(S) À COMPLÉTER

        fic.close();
        fic.clear();

        if (nbArtistes > 0)
        {
            estLu = true;

            //4.2 (0,75 point) Création du tableau dynamique à
            //l'aide du paramètre artistes

            INSTRUCTION(S) À COMPLÉTER

            fic.open(nomFichier.c_str());

            //4.3 (3,5 points) Lecture du fichier et mémorisation
            //de l'information dans le tableau dynamique

            INSTRUCTION(S) À COMPLÉTER

            fic.close();
        } // fin de if (nbArtistes >0)
    } // fin de if ( !fic.fail())

    return estLu;
}
```