

### Exercise 1 ( 20 pts )

We say that a word S1 is a *left factor* of a word S2 if and only if S2 begins with the characters making up the word S1. For example,

The word « poly » is a left factor of the word « polycopie », while

The word « jour » is not a left factor of the word « bonjour »

Use C# language to write a program that reads two words S1 and S2 and then tests whether S1 is a *left factor* of S2.

### Exercise 2 ( 80 pts )

Read all the points carefully before answering the questions.

Let us consider the web application *Exam* having the WebForm « **Inscription.aspx** »

## Inscription

LastName followed by firstName :	<input type="text"/>	Sexe :	<input type="text"/>																																																	
<i>Name can not be empty</i>			<i>Sexe should be "F" or "M"</i>																																																	
Age :	<input type="text"/>	Code 1 :	<input type="text" value="I3332"/>																																																	
<i>Age should be &gt;= 18</i>		Code 2 :	<input type="text"/>																																																	
			<i>Code 2 should be the same of Code 1</i>																																																	
Inscription Date :	<div style="text-align: center;"> <div>&lt; July 2020 &gt;</div> <table border="1"> <thead> <tr> <th>Sun</th><th>Mon</th><th>Tue</th><th>Wed</th><th>Thu</th><th>Fri</th><th>Sat</th></tr> </thead> <tbody> <tr> <td>28</td><td>29</td><td>30</td><td>1</td><td>2</td><td>3</td><td>4</td></tr> <tr> <td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td></tr> <tr> <td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td></tr> <tr> <td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td></tr> <tr> <td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td><td>1</td></tr> <tr> <td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td></tr> </tbody> </table> </div>			Sun	Mon	Tue	Wed	Thu	Fri	Sat	28	29	30	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8
Sun	Mon	Tue	Wed	Thu	Fri	Sat																																														
28	29	30	1	2	3	4																																														
5	6	7	8	9	10	11																																														
12	13	14	15	16	17	18																																														
19	20	21	22	23	24	25																																														
26	27	28	29	30	31	1																																														
2	3	4	5	6	7	8																																														
			<i>Selected Date</i>																																																	
			<i>nb of days remainder</i>																																																	
<input type="button" value="Add"/>																																																				

### Summary :

Unbound	<input type="button" value="Delete"/>	<input type="button" value="DeleteAll"/>
---------	---------------------------------------	--

- The student's name is required (i.e., cannot be empty). It is given, in the **txtName** textbox, such as the family name is followed by the student's first name (example: Naja Nada);
- The sex, given in the **txtSexe** textbox, can only have the two values "F" or "M";

- The age, is given in the **txtAge** textbox, and its value must be greater than or equal to 18;
- Code 1, is given in the **txtCode1** textbox and, contains the fixed value **I3332** which cannot be modified during the execution of the application;
- Code 2, is given in the **txtCode2** textbox, and must have the same value as code 1;
- The inscription date is chosen by the **Calendar1** calendar;
- The button Add **btnAdd** allows :
  - a) To test the selected inscription date of Calendar1;
    - If this date is lower than the system date, then an error message is displayed in label **labInscrip** as well as in label **labnbDay**;
    - otherwise, the value of the selected date is displayed in the **labInscrip** label, and a message specifying the number of days remaining for the exam will be displayed in the **labnbDay** label (for example: *your exam is in 3 days*).
  - b) Add, if the selected date is greater than or equal to the system date, in the **listInscrip** listbox of the Summary part, a string containing the student's first name followed by his family name followed by "having" followed by his age followed by "registered in" followed by the selected date (for example: *Nada Naja having 23 years registered in 27-Jul-20*)
- The Delete button **btnDelete** allows to delete the selected item from the **listInscrip** listbox.
- The Delete All button **btnDeleteAll** allows to empty the listbox **listInscrip**.

**Q1.** Write the script of buttons Add , Delete and Delete All.

**Q2.** What property should be used for the **txtCode1** textbox to prevent the user from modifying its value during the execution of the application? Give the necessary value of this property

**Q3.** What is the value of the regular expression (i.e., ValidationExpression) that should be used for the **txtSexe** textbox allowing to reject any value different from *F* or *M*?

**Q4.** Which validation control should be used to verify that the student's age will be greater than or equal to 18? Write the values of the properties necessary for this validation control.

**Q5.** Which validation control should be used to ensure that the **txtName** textbox is not leaved empty? Write the values of the properties necessary for this validation control.

**Q6.** Which validation control should be used to ensure that the value of code 2 is the same as that of code 1? Write the values of the properties necessary for this validation control.

Examples of execution (Note that the printing date of exam is 17 July 2020) :

## Inscription

LastName followed by firstName :  Sexe :   
Age :  Code 1 :   
Code 2 :

Inscription Date :

July 2020						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
28	29	30	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1
2	3	4	5	6	7	8

27-Jul-20

Your exam is in 10 days

Add

## Summary :

Nada Naja having 23 years registered in 27-Jul-20  
Marwan Masri having 28 years registered in 30-Jul-20

Delete

DeleteAll

## Inscription

LastName followed by firstName :  Sexe :   
Age :  Code 1 :   
Code 2 :

Inscription Date :

July 2020						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
28	29	30	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1
2	3	4	5	6	7	8

Error !

selectd day can not preceed current day !

Add

## Summary :

Nada Naja having 23 years registered in 27-Jul-20  
Marwan Masri having 28 years registered in 30-Jul-20

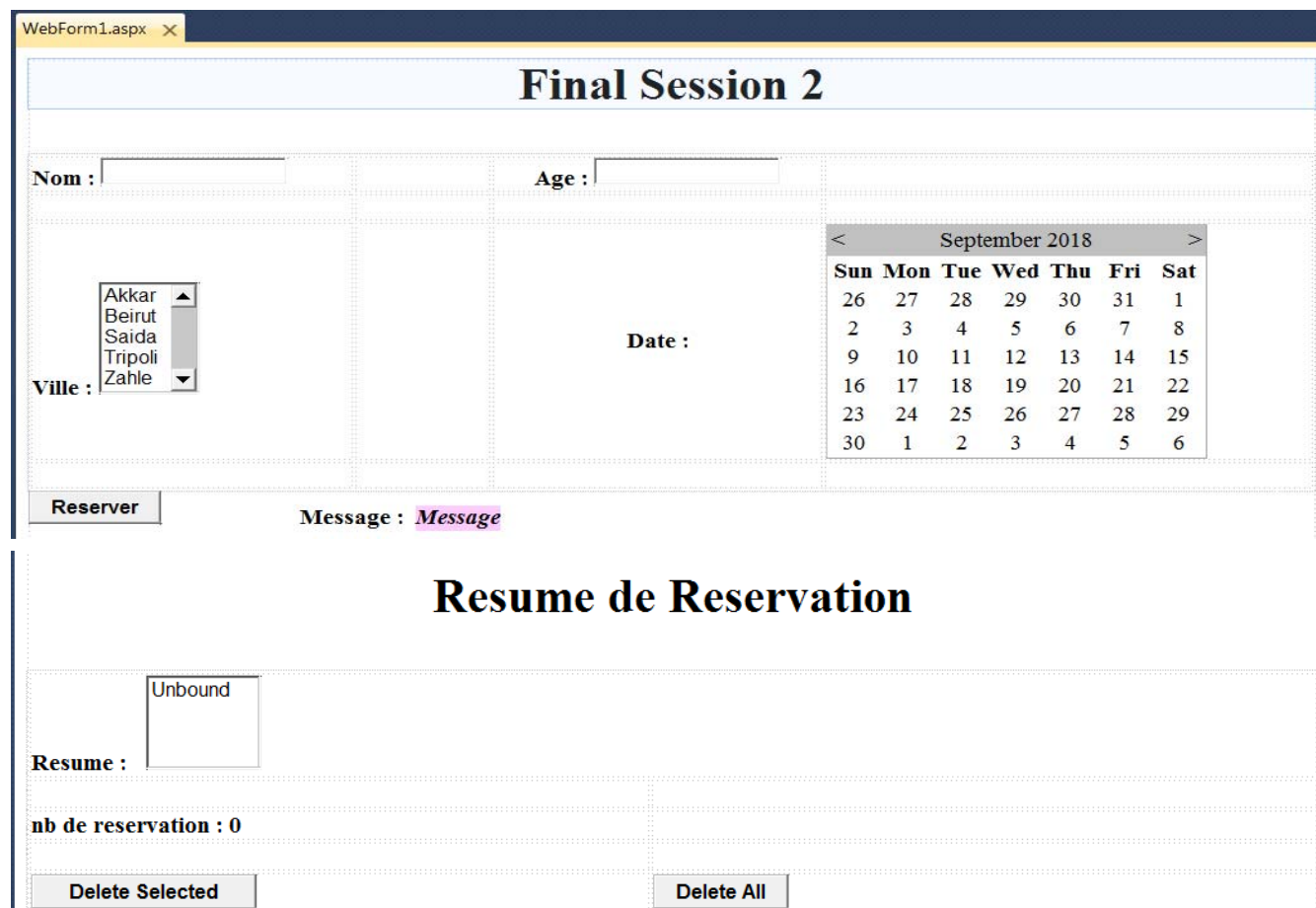
Delete

DeleteAll

**Good Luck**

## Exercice 1 ( 60 pts)

A - Soit l'application web *Session2Final* contenant le WebForm « **WebForm1.aspx** ».



The screenshot shows a web application titled "Final Session 2". It contains a form with the following elements:

- Nom :** A text input field.
- Age :** A text input field.
- Ville :** A dropdown menu with options: Akkar, Beirut, Saida, Tripoli, Zahle.
- Date :** A calendar control showing September 2018.
- Reserver** button.
- Message :** A label displaying the message "Message".
- Resume de Reservation** section with an "Unbound" data source.
- nb de reservation :** A label displaying "0".
- Delete Selected** and **Delete All** buttons.

Ecrire le script correspondant aux boutons « *Reserver* », « *Delete Selected* » et « *Delete All* » tel que :

- Le bouton « *Reserver* » (***btnReserver***)
  - ❖ permet d'abord d'ajouter, dans la ListBox ***listResume***, la réservation faite sous forme de :  
" *nomX* , ayant *ageX* ans, habitant a *villeX* , a reserver le *dateX* " ; Par exemple : " *Alia Haddad*, ayant 41 ans, habitant a *Tripoli*, a reserver le 15-Sept-18 " ;
  - ❖ vide ensuite les TextBox de nom (***txtNom***) et age (***txtAge***),
  - ❖ puis remet le curseur sur la 1<sup>ère</sup> ville dans la ListBox ***listVille*** et la date dans le calendrier ***Calender1*** a la date d'aujourd'hui.
  - ❖ permet également d'afficher un message (à côté du bouton réserver) dans le label ***labelMessage*** tel que : si le date de réservation sélectionnée est la date d'aujourd'hui, alors

le message affiché est « *Bon Voyage !* » ; sinon le message affiché est « *Il reste nbX jours pour votre evenement !* » ; Par exemple, pour Alia Haddad qui a réservé pour le 15-Sept-18, le message affiché sera : « *Il reste 3 jours pour votre evenement !* » ;

- Le bouton « Delete Selected » (***btnDelSel***) permet de supprimer de la liste, la réservation sélectionné.
- Le bouton « DeleteAll » (***btnDelAll***) permet de vider la liste des réservations.
- Le nombre des réservations est un label « ***labelNb*** », initialisé à 0, et qui sera incrémenté (resp. décrémenté) à chaque ajout (resp. suppression) d'une réservation dans (resp. de) la liste.

**B** – On modifie les propriétés concernant le textbox *txtNom* tel que le code asp devient le suivant :

```
<asp:TextBox ID = "txtNom" runat = "server" ForeColor = "Blue"  
    TextMode = "MultiLine" > </asp:TextBox>
```

Que signifie chacune de ces propriétés ?

## **Exercice 2** ( 40 pts )

**A** – En utilisant le langage C# , créer une classe *Animal* contenant les attributs et les méthodes suivants :

- *nbPattes* qui représente le nombre des pattes de l'animal ;
- *num* qui représente le numéro de l'animal (nombre entier) ;
- *age* qui représente l'âge de l'animal (nombre entier) ;
- *nbAnimal* qui correspond au nombre des animaux créés.
- Un constructeur permettant d'initialiser les attributs par des valeurs fournit en paramètres ;
- Un constructeur sans paramètres permettant d'initialiser les attributs à 0 en faisant appel au premier constructeur ;
- Un constructeur avec un paramètre *Animal* permettant d'initialiser les attributs ;
- Une méthode *getNb()* qui retourne le nombre des objets créés de la classe *Animal*.
- Une méthode *afficheAnimal()* permettant d'afficher les caractéristiques de la classe *Animal*.

**B** - Créer une classe *Oiseau* qui dérive de la classe *Animal*. Elle possède une couleur du bec. Définir pour cette classe :

- Un constructeur permettant d'initialiser les attributs de la classe par les valeurs fournit en paramètres, en utilisant le premier constructeur de la classe *Animal*.
- Une méthode *afficheOiseau()* permettant d'afficher les caractéristiques de la classe *Oiseau*.

**C.** Implémenter une méthode *main* pour afficher les caractéristiques de l'animal **A1**(4, 123, 3) et les caractéristiques de l'oiseau **O1**(2, 555, 1, "black") ; Ensuite, afficher le nombre des animaux créés.

**Examen TP : Mercredi 19 Sept à 9h**

**Partie 1 : ( 70 pts )**

Soit l'application web **Examen** contenant le WebForm « **Conge.aspx** » .

Conge.aspx X

## Holidays application :

Full name Employee :  Sex :

Name is Obligatory

Phone number :  Reason :

should be as XX-XX-XX-XX Label reason sex

Starting date : 

Jul	August 2019						Sep
Sun	Mon	Tue	Wed	Thu	Fri	Sat	
28	29	30	31	1	2	3	
4	5	6	7	8	9	10	
11	12	13	14	15	16	17	
18	19	20	21	22	23	24	
25	26	27	28	29	30	31	
1	2	3	4	5	6	7	

 End date : 

Jul	August 2019						Sep
Sun	Mon	Tue	Wed	Thu	Fri	Sat	
28	29	30	31	1	2	3	
4	5	6	7	8	9	10	
11	12	13	14	15	16	17	
18	19	20	21	22	23	24	
25	26	27	28	29	30	31	
1	2	3	4	5	6	7	

Label Start date Label End date

Label Message

Validate Cancel Apply

List of Employees :  Delete Selected Delete All

Nb of Employees : 0

**Remarque :** Un exemple d'exécution est ajouté dans la page 4.

- Le nom de l'employé est saisi, dans le textbox *txtName* ;
- Le sexe de l'employé est un dropDownList *ddlSex* , qui a les valeurs F et M ;
- Le téléphone de l'employé est saisi, dans le textbox *txtPhone* ;
- La raison pour le congé est un dropDownList *ddlReason* , qui a les valeurs : « Sick » , « Maternity » , « Wedding » , « Death » ;
- La date du début et de fin de congé sont resp. choisi des calendrier *CalendarStart* et *CalendarEnd* ;

**Q1.** Ecrire le script du bouton Valider **btnValider** permettant

- a) de rejeter la sélection dans le calendrier **CalendarStart** d'une date précédant la date du système. Le message d'erreur doit être écrit dans le label **LabStartDate**. Si la date sélectionnée est correcte alors la valeur de **LabStartDate** sera le message « *From* » suivi par la date sélectionnée. Par exemple : *From 12-Jul-19* ;
- b) de rejeter la sélection dans le calendrier **CalendarEnd** d'une date précédant la date du sélectionnée dans le calendrier **CalendarStart** . Le message d'erreur doit être écrit dans le label **LabEndDate**. Si la date sélectionnée est correcte alors la valeur de **LabEndDate** sera le message « *To* » suivi par la date sélectionnée. Par exemple : *To 31-Jul-19* ;
- c) d'interdire un homme de choisir un congé de maternité. Le message d'erreur doit être écrit dans le label **LabReasonSex** (se trouvant au-dessous du dropdownlist *ddlReason*). Si la raison sélectionnée est compatible avec le sexe choisit alors la valeur de **LabReasonSex** sera vide.

**Q2.** Ecrire le script du bouton Cancel **btnCancel** permettant de

- a) vider les textbox *txtName* et *txtPhone* ainsi que tous les labels du webForm ;
- b) remettre la date dans les 2 calendriers à la date d'aujourd'hui
- c) remettre le curseur sur le 1<sup>er</sup> élément dans les 2 dropdownlist.

**Q3.** Ecrire le script du bouton Apply **btnApply** permettant

- a) d'afficher dans le label **LabMessage** un message contenant le préfix « *Mr* » ou « *Mme* », selon le sexe de l'employé, suivi par le nom de l'employé suivi par un message précisant le nombre de jours restant pour le début de vacances. Par exemple, *Mr Sami Osman, your holidays will be in 3 days* ;
- b) d'ajouter dans la liste des employées **listEmpl** le nom de l'employé suivi par le nb de jours de vacances qu'il a sélectionné.
- c) d'afficher le nombre des employés ayant pris un congé dans le label **labnb** ;

**Q4.** Ecrire le script du bouton DeleteSelected **btnDelSel** permettant de supprimer, de la liste des employés, la ligne sélectionnée. N'oubliez pas de mettre à jour la valeur du label **labnb**.

**Q5.** Ecrire le script du bouton DeleteAll **btnDelAll** permettant de vider la liste des employées et de rendre le nombre des employés à 0 dans le label **labnb**;

**Q6.** Quelle est la différence entre un évènement **Load** et un évènement **Init** d'une page ASP.net ?  
Lequel de ces deux évènements se produit avant ?

**Q7.** On souhaite affecter la valeur « -1 » au nombre des employés dans le label **labnb** lorsque la page est chargée la première fois par le serveur web. Faut-il initialiser cette valeur par un script concernant l'évènement *Load* ou l'évènement *Init* ? Donnez le script correspondant.

**Q8.** a) Quel type de control de validation faut-il utiliser pour obliger l'utilisateur de saisir une valeur dans le textbox **txtName** (c.à.d. ne pas accepter le vide dans le textbox) ? Donnez les valeurs des propriétés suivantes de ce control de validation : i) *ControlToValidate* ii) *Display*

b) A quoi sert la propriété *ForeColor* ?

**Q9.** Quelle est la valeur de l'expression régulière (i.e., *ValidationExpression*) qu'il faut utiliser pour le textbox **txtPhone** pour obliger l'utilisateur à saisir une valeur de la forme xx-xx-xx-xx (où x est un chiffre quelconque) ? Par exemple, 71-33-33-33.

**Q10.** Si on modifie les propriétés du textbox **txtName** tel que le code devient le suivant :

```
<asp:TextBox ID="txtName" runat="server" MaxLength="20"
    TextMode="Password" ToolTip="First and Last name ">
</asp:TextBox>
```

Que Signifie chacune des propriétés précédentes ?

**Q11.** On ajoute un textbox **txtAge** pour insérer l'âge de l'employé ; on ajoute également un control de validation pour rejeter une valeur inférieure à 23 ans pour l'âge. Donnez les valeurs (si besoin, sinon expliquez pourquoi il faut laisser vide) des propriétés suivantes du control de validation :

a) *ControlToCompare*      b) *ControlToValidate*      c) *ValueToCompare*      d) *Operator*

## **Partie 2 :**                      **( 30 pts )**

Un nombre complexe est de la forme  $X = a + i.b$  où  $a$  est la partie réelle et  $b$  est la partie imaginaire. Soient  $X$  et  $Y$  deux nombres complexes tel que  $X = a + i.b$  et  $Y = c + i.d$ . L'addition, la soustraction et la multiplication de deux nombres complexes est un nombre complexe  $Z$  tel que :

$$Z = X + Y = (a + c) + i.(b + d)$$

$$Z = X - Y = (a - c) + i.(b - d)$$

$$Z = X * Y = (a*c - b*d) + i.(a*d + c*b).$$

1) Utilisez le langage C# pour créer une classe *NombreComplexe* contenant les attributs :

- $a$  de type réel pour représenter la partie réelle ;
- $b$  de type réel pour représenter la partie imaginaire ;

2) Créez un constructeur avec paramètres qui permet d'initialiser les attributs de la classe.



- 3) Créez un constructeur sans paramètres qui permet d'initialiser les attributs de la classe à 0 ;
- 4) Créez les fonctions de modifications et d'accès aux deux champs *a* et *b* (i.e. méthodes *set* et *get*).
- 5) Créez une méthode pour l'addition, une méthode pour la soustraction, et une méthode pour la multiplication de deux nombres complexes.
- 6) Créez une méthode pour l'affichage d'un nombre complexe sous forme de  $a + i.b$  ;
- 7) Implémentez la méthode *main* dans laquelle l'utilisateur saisi les champs des deux nombres complexes X1 et X2 puis affiche le résultat de l'addition, soustraction et multiplication de ces deux nombres.

### Exemple d'exécution de la Partie 1:

**Remarque** : date de l'impression de l'examen (i.e., date du système) est le 30 aout 2019 ;

## Holidays application :

Full name Employee : 
Sex :

Phone number : 
Reason :

Starting date :

Aug	September 2019						Oct
Sun	Mon	Tue	Wed	Thu	Fri	Sat	
25	26	27	28	29	30	31	
1	2	3	4	5	6	7	
8	9	10	11	12	13	14	
15	16	17	18	19	20	21	
22	23	24	25	26	27	28	
29	30	1	2	3	4	5	

From 02-Sep-19

End date :

Aug	September 2019						Oct
Sun	Mon	Tue	Wed	Thu	Fri	Sat	
25	26	27	28	29	30	31	
1	2	3	4	5	6	7	
8	9	10	11	12	13	14	
15	16	17	18	19	20	21	
22	23	24	25	26	27	28	
29	30	1	2	3	4	5	

To 07-Sep-19

Mr Thomas Dubois, your holidays will be in 3 days

List of Employees :

Sami Osman, 2 days

Thomas Dubois, 5 days

Nb of Employees :

2

**Bonne Chance**

### Exercise 1 ( 20 pts )

Write a C# program that reads an integer number N (N should be between 3 and 9), then display on the screen a line composed of N characters "+" followed each by a character "-".

*Example of execution:*

```
give n between 3 and 9 : 12
give n between 3 and 9 : 6

your graph is : + - + - + - + - + - + -
```

### Exercise 2 ( 80 pts )

Read carefully all the points with the execution examples before answering the questions.

Let us consider the web application **Exam** having the WebForm « **Inscription.aspx** »

Name :	<input type="text"/>	Name can not be empty	Phone :	<input type="text"/>																																																	
Sexe :	<input type="text"/>	Sexe should be F or M	Phone should be as xxx-xx-xxx-xxx																																																		
Date :	<div> <div>&lt;</div> <div>July 2020</div> <div>&gt;</div> </div> <table border="1"> <tr> <td>Sun</td><td>Mon</td><td>Tue</td><td>Wed</td><td>Thu</td><td>Fri</td><td>Sat</td> </tr> <tr> <td>28</td><td>29</td><td>30</td><td>1</td><td>2</td><td>3</td><td>4</td> </tr> <tr> <td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td> </tr> <tr> <td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td> </tr> <tr> <td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td> </tr> <tr> <td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td><td>1</td> </tr> <tr> <td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td> </tr> </table>	Sun	Mon	Tue	Wed	Thu	Fri	Sat	28	29	30	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	<p>Selected Date</p> <p>nb of days remainder</p>	Grade 1 :	<input type="text"/>
Sun	Mon	Tue	Wed	Thu	Fri	Sat																																															
28	29	30	1	2	3	4																																															
5	6	7	8	9	10	11																																															
12	13	14	15	16	17	18																																															
19	20	21	22	23	24	25																																															
26	27	28	29	30	31	1																																															
2	3	4	5	6	7	8																																															
			Grade 2 :	<input type="text"/>																																																	
		Validate	Grade 2 should be >= Grade 1																																																		
Summary :																																																					
Unbound		Delete	DeleteAll	Nb of Student :	0																																																

- The student's name is required (i.e., cannot be empty). It is given, in the **txtName** textbox, such as the family name is followed by the student's first name (example: *Naja Nada*);
- The sex, given in the **txtSexe** textbox, can only have the two values "F" or "M";
- Phone, given in textbox **txtPhone**, should be of the form xxx-xx-xxx-xx where x is any number (example : 961-01-123-123) ;

- The 2 grades, given respectively in the textboxes **txtGrade1** and **txtGrde2**, such that value of Grade 2 should be greater than or equal to Grade 1 ;
- The inscription date is chosen by the **Calendar1** calendar;
- The button Add **btnAdd** allows :
  - a) To test if the selected inscription date of **Calendar1** is valid;
    - If this date is lower than the system date, then an error message is displayed in label **labInscrip** as well as in label **labnbDay**;
    - otherwise, the value of the selected date is displayed in the **labInscrip** label, and a message specifying the number of days remaining for the exam will be displayed in the **labnbDay** label (for example: *your exam is in 3 days*).
  - b) Add, if the selected date is greater than or equal to the system date, in the **listInscrip** listbox of the Summary part, a string containing the prefix “Mr. ” or “Mme.” according to the sex of the student, followed by the student's first name followed by his family name , followed by the word " get " followed by the average of grades that this student get in the 2 exams, followed by “ as average grade “ (for example: *Mr. Fuad Fattal get 45.5 as average garde*); Then display, in the label **labNbStud**, the number of student added in the list **listInscrip** .
- The Delete button **btnDelete** allows to delete the selected item from the listbox **listInscrip** and update the value of the number of student in **labNbStud**.
- The Delete All button **btnDeleteAll** allows to empty the listbox **listInscrip** and resets to zero the number of students in **labNbStud**.

**Q1.** Write the script of buttons Validate , Delete and Delete All.

**Q2.** What is the value of the regular expression (i.e., ValidationExpression) that should be used for the textbox **txtPhone** to reject any value that does not match the requested value?

**Q3.** Which validation control should be used for the textbox **txtSexe** allowing to reject any value other than *F* or *M*? Write the values of the properties necessary for this validation control

**Q4.** Which validation control should be used on **txtGrade2** to verify that value of Grade 2 is  $\geq$  to that of Grade 1? Write the values of the properties necessary for this validation control.

**Q5.** Which validation control should be used to ensure that the **txtName** textbox is not leaved empty? Write the values of the properties necessary for this validation control.

**Examples of execution (Note that the printing date of exam is 29 Sept 2020) :**

Name :  Phone :

Sexe :

Date : 

September 2020						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	1	2	3
4	5	6	7	8	9	10

10-Oct-20  
Your exam is in 11 days

Grade 1 :

Grade 2 :

**Summary :**

Mr . Fuad Fattal get 43 as average grade  
Mme . Nada Naja get 52 as average grade  
Mr . Marwan Masri get 63 as average grade

Nb of Student : 3

Name :  Name can not be empty

Sexe :  Sexe should be F or M

Date : 

September 2020						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	1	2	3
4	5	6	7	8	9	10

Error !  
selectd day can not preceed current day !

Grade 1 :

Grade 2 :

Grade 2 should be >= Grade 1

**Good Luck**

**Exercise 1 : ( 80 pts )**

Consider the web application **Exam** having 2 web form « **Information.aspx** » and « **Summary.aspx** ».

Summary.aspx Information.aspx X

Name (first name followed by last name) :  Sexe : ☐ F ☐ M  
**Obligatory field**

Phone number :   
**should be as +XXX-XX-XXX-XXX**

Age :   
**Age should be 18 and 25**

Date Exam 1 : 

June 2021						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	1	2	3
4	5	6	7	8	9	10

**date of Exam 1 should be >= current date**

Date Exam 2 : 

June 2021						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	1	2	3
4	5	6	7	8	9	10

**date Exam 2 should be >= date Exam 1**

Grade Exam 1 :  **Grade should be between 0 and 100**

Grade Exam 2 :  **Grade should be between 0 and 100**  
**Grade 2 should be >= Grade 1**

Final Grade :

List of Students :

Summary.aspx X Information.aspx

body

Mr / Ms , Last Name

Grade 1 :  Grade 2 :

Final Grade :  Failed / Succeed

In the web form « **Information.aspx** » :

- Student's name, is given in the textbox *tbName*, such as the first name is followed by family name;
- Student's gender (sex) is a radio button list *rbListSexe*;
- Student's phone is given in the textbox *tbPhone* ;
- Exam dates are resp. chosen from calendars *CalExam1* and *CalExam2*;
- Exam grades are resp. given in textbox *tbGrade1* and *tbGrade2* ;
- Final exam grade should be calculated in the label *labGrade* (30% for Exam1 and 70% for Exam2);

**Q1.** How to set the page « Information.aspx » as previous page of the web form « Summary.aspx » ?

**Q2.** Write the script of button Validate **btnValidate** allowing

- a) To reject the selection in the calendar **CalExam1** of a date preceding the system date . The error message should be written in the label **labDateExam1**. If the selected date is correct then the value of **labDateExam1** will be the message « Exam 1 : » followed by the selected date. For Example : *Exam 1 : 12-Jul-21 ;*
- b) To reject the selection in the calendar **CalExam2** of a date preceding the selected date of **CalExam1**. The error message should be written in the label **labDateExam2**. If the selected date is correct then the value of **labDateExam2** will be the message « Exam 2 : » followed by the selected date. For example : *Exam 2 : 22-Jul-21 ;*
- c) To calculate the final grade knowing that exam 1 has 30% and exam 2 has 70% ;
- d) To add in the list box **listStud** the student's name followed by his final grade followed by the word « Succeed » or « Failed » depending on whether his final grade is greater or less than 50 and such that, failed students are at the bottom of the list and successful students are at the top of list ;

**Q3.** Write the script of the button Clear **btnClear** allowing to

- a) Empty all text boxes as well as labels of the web form « Information.aspx » ;
- b) Put the selected date of both calendars to today's date;
- c) Not select any sex value.

**Q4.** Write the script of button Delete **btnDelete** allowing to remove the selected item from the list **listStud**;

**Q5.** Write the script of the button UP **btnUP** allowing to move up the selected item of the list **listStud**.

**Q6.** Write the script of the button Down **btnDown** allowing to move down the selected item of the list **listStud**.

**Q7.** Write a script allowing to add a property **propertySexe** to retrieve the selected value of the radio button list **rbListSexe** ;

**Q8.** Write the script allowing to display the values of fields in the web form **Summary.aspx** (**tbNameSexe**, **txtGrade1**, **txtGrade2**, **txtFinal**), retrieved from the page **Information.aspx**. In **tbNameSexe**, we display « Mr » or « Ms » depending on the student's sex, followed by the student's family name. The result « Failed » or « Succeed » is displayed in the label **labResutl** depending on whether the final grade is greater or smaller than 50 ;

**Q9.** We want to assign the value « -1 » to the final grade in the label *labGrade* of the page *Information.aspx* when the page is loaded for the first time by the web server. Should this value be initialized by a script concerning the *Load* event or *Init* event ? Write the corresponding script.

**Q10.** Which property should be used to display a message when the mouse is over the text box *tbName* ?

**Q11.** What is the value of the regular expression (i.e., *ValidationExpression*) that should be used for the textbox *tbPhone* to oblige the user to give a value of the form +XXX-XX-XXX-XXX (where x is any digit) ? For example, +961-01-123-456.

**Q12.** What does each of the following properties of the textbox *tbTest* means ?

ID="tbTest"      runat="server"      MaxLength="21"      TextMode="Password"

**Q13.** Which validation control should be used to oblige the user to choose a sex? Write the necessary properties for this validation control (as well as the values of these properties).

**Q14.** Which validation control should be used to verify that the given value of age is between 18 and 25 ? Write the necessary properties for this validation control (as well as the values of these properties).

**Q15.** Which property should be used to display the value « 0 » in the textbox *tbGrade1* when the web page is loaded for the first time ?

**Q16.** . Which validation control should be used to verify that the given value of *tbGrade2* is greater than that of *tbGrade1* ? Write the necessary properties for this validation control (as well as the values of these properties).

**Q17.** Write the necessary steps allowing to rename « *Information.aspx* » to « *Personal.aspx* ».

## **Exercise 2 :      ( 20 pts )**

Write a program in which a constant x is defined as a secret integer. The program should then ask the user to guess this secret number by giving integer numbers until the given number equals x, by indicating each time whether the number given by the user is less or greater than x. Example of execution of the program for x = 12 :

```
Guess the secret number : 5
Try a bigger number !
Guess the secret number : 45
Try a smaller number !
Guess the secret number : 12
Excellent ! you guessed the secret number !
```

***Good Luck***

**Exercise 1 : ( 75 pts )**

Create an ASP.Net application that allows the validation of a form for a credit card information. Data should respect the following constraints:

- ↳ All fields in the « *Credit Card Details* » section are obligatory
- ↳ The expiry date should be greater than current date and should be MM/YYYY format such that MM is the month and YYYY is the year (Attention: 13/2021 for example is invalid!)
- ↳ The card key is a 4 digits integer number
- ↳ The amount should be a positive integer between 100 and 10000
- ↳ The card number is an integer that should be valid according to the following rule:

The card number has a length (number of digits) and a numerical prefix according to its type

Card Type	Prefix	Number of digits	Example
MasterCard	from 51 to 55	16	5309 2101 7304 9302
Visa	4	13 or 16	4012 8291 7392 1881
American Express	34 or 37	15	3714 4963 5398 431

- The name of the cardholder is given in the textbox **tbName**, such that the first name is followed by the family name
- The card type is a radio button list **rbCardType**
- The card number is given in the text box **tbCardNum**
- The expiry date of the card is given in the text box **tbCardExpiry**
- The card key is given in the text box **tbCardKey**
- The amount to pay by the card is given in the text box **tbAmount**
- The total sum to pay should be calculated in the text box **tbSum**
- The total number of purchase operations should be calculated in the label **labTotalNb**
- The list **shoppinglist** is a list box that contain the purchase operations of the store

**Q1.** Write the script for the button Delete **btnDelete**, allowing to remove the selected item from the list box **shoppinglist** ; Don't forget to update the total sum to be paid (which is the sum of all amounts to pay by all customers of the store) as well as the total number of purchase operations.

**Q2.** Write the script for the button DeleteALL **btnDeleteALL**, allowing to empty the list box **shoppinglist** ; Don't forget to update the total sum to be paid as well as the total number of purchase operations.



**Q3.** Write the script for the button Clear **btnClear** allowing to

- a) Empty all fields of the « *Credit Card Information* » section;
- b) Not select any card type value.

**Q4.** Write the script for the button ADD **btnADD** allowing to

- a) add to the list box *shoppinglist* the cardholder's name (such that the family name is followed by first name separated by a space) followed by a semicolon followed by the amount to be paid ;
- b) add the amount to be paid to the total amount to be paid ;
- c) increment the total number of purchase operations ;

**Q5.** Write the script of the button Validate **btnValidate** allowing to check the constraints concerning the validity of the card (card number and its expiry date).

**Q6.** What is the difference between a **Load** event and an **Init** event on an ASP.net page? Which of these two events occurs before?

**Q7.** Which property should be used to display the value « 0 » in the label *labTotalNb* when the web page is loaded for the first time ?

**Q8.** Which property should be used to prevent the user from entering values in *tbSum* ?

**Q9.** Which validation control should be used to reject any incorrect card key value? Write the necessary properties for this validation control (as well as the values of these properties).

**Q10.** Which validation control should be used to reject any incorrect value for the amount to be paid ? Write the necessary properties for this validation control (as well as the values of these properties).

**Q11.** Which validation control should be used to reject the empty value in the text box *tbName* ?

**Q12.** Which validation control should be used to reject any value, of the card number, that is not an integer number? Write the necessary properties for this validation control (as well as the values of these properties).

**Q13.** Which property should be used for the list box *shoppinglist* in order to be able to select several items from the list box?

**Q14.** Which property should be used for the text box *tbName* to prevent the user from entering a name with more than 30 characters ?

**Q15.** Which property should be used, for the credit card type *rbICardType*, in order to select the direction (Horizontal or Vertical) in which the elements must be arranged ?

**Q16.** Which validation control should be used to reject a value that is not in the form MM/YYYY of the credit card expiry date ? Write the necessary properties for this validation control (as well as the values of these properties).

Credit Card Details	
Type of Card :	<input type="radio"/> MasterCard <input type="radio"/> Visa <input type="radio"/> American Express
Card Number :	<input type="text"/>
Expiry Date :	<input type="text"/>
Key :	<input type="text"/>
Card Holder : (1st name last name)	<input type="text"/>
Amount :	<input type="text"/>
<input type="button" value="Validate"/>	<input type="button" value="Add"/> <input type="button" value="Clear"/>

Store Shopping List	
<input type="text" value="Unbound"/>	
<input type="button" value="Delete"/>	<input type="button" value="Delete ALL"/>
Sum to pay :	<input type="text" value="0"/>
nb of purchase :	<input type="text" value="0"/>

## **Exercise 2 :** ( 25 pts )

Write a program that asks the user to give an integer number *n* that should be composed of four digits (i.e.,  $999 < n < 9999$ ); The program should then display its mirror.

For example, if *n* = **2835** then its mirror is **5382**

***Good Luck***

**Exercice 1 ( 60 pts )**

Soit l'application web **Examen** contenant le WebForm « **WFPartiel.aspx** ».

1 - Ecrire le script correspondant aux boutons « *Clear* » et « *Insert* » sachant que :



- Le bouton « *Clear* » (ID = **btnClear**) permet de vider les TextBox de nom (ID = **tbNom**) , age (ID = **tbAge**) et note (ID = **tbNote**).
- Le bouton « *Insert* » (ID = **btnInsert**) permet d'ajouter le nom de l'étudiant ainsi que son âge et sa note dans la ListBox *Resume* (ID = **lbResume**) sous la forme de : *nomX* age de *ageX* ans a eu la note *noteX* ; Par exemple : *Sami* age de *21* ans a eu la note *67*

2 - Que signifie chacune des propriétés suivantes concernant le control de validation :

```
<asp:RangeValidator ID="RangeValidator1" runat="server" ControlToValidate="tbNote"
```

```
Display="Dynamic" ErrorMessage="note doit etre entre 0 et 100" ForeColor="Red"
```

```
MaximumValue="100" MinimumValue="0" Type="Double"> </asp:RangeValidator>
```

*Tag runat="server" indicates that the code contained within the script block will run on the server (and not on the client). On execution, ASP.NET will create server-side objects that contain this code as well as an instance of the Page class to contain the controls defined inside the page as instances of their given type (System.Web.UI.WebControls.Textbox, for example). This server-side object will be invoked on user request and will execute code in response to events.*

3 - Donnez les valeurs (si besoin, sinon expliquez pourquoi il faut laisser vide) des propriétés

suyvantes qui manquent, pour le control de validation, afin de contrôler que la valeur du textbox age (**tbAge**) soit un nombre entier :

```
<asp:CompareValidator ID="CompareValidator1" runat="server"
    ControlToValidate="TextBox2" Display="Dynamic"
    ErrorMessage="Should be integer !" ForeColor="Red" Operator="DataTypeCheck"
    style="font-style: italic" Type="Integer"></asp:CompareValidator>
```

```
<asp:CompareValidator ID="CompareValidator1" runat="server"
    ControlToCompare=" ?????? " ControlToValidate=" ?????? " Display="Dynamic"
    ErrorMessage="Valeur de Age doit etre entier" ForeColor="Red" Operator=" ?????? "
    Type=" ?????? " > </asp:CompareValidator>
```

3 - On souhaite affecter la valeur « -1 » à la note de l'étudiant lorsque la page est chargée la première fois par le serveur web.

Faut-il initialiser cette valeur par un script concernant l'évènement *Load* ou l'évènement *Init* ?

Donnez le script correspondant.

```
protected void Page_init(object sender, EventArgs e)
{
    TextBox3.Text = "-1";
}
```

## **Exercice 2** ( 40 pts )

1 - Utilisez le langage C# pour créer une classe **Nombre** contenant :

- Un champ entier x.
- Une méthode *afficheNb* permettant d'afficher la valeur du champ x.
- Une méthode *sommeDiviseur* qui retourne la somme des diviseurs entiers du champ x (1 est compris mais x non compris). Exemple si x = 6, alors somme des diviseurs de 6 = 1 + 2 + 3 = 6
- Une méthode *amis* qui teste si la valeur du champ d'un nombre est ami avec la valeur du champ x. Deux entiers sont dits amis si la somme des diviseurs de l'un est égale à l'autre et vice-versa.
- Une méthode *nombreAmis* qui affiche les valeurs des champs des nombres amis de x, inférieure à 2000.

2 - Implémentez la méthode *main* dans laquelle l'utilisateur saisi la valeur du champ d'un nombre puis affiche la valeur du champ de son nombre ami.

***Bonne Chance***

مطلوب الإجابة على كل جزء بشكل متواصل ودون تداخل مع الجزء الآخر

## Partie Partiel (~ 45 min)

### Exercice 1 ( 100 pts) : A(43pts) ; B(32pts) ; C (25pts)

Utiliser le langage C# dans les trois parties A, B et C.

**A.** Créer une classe **Bateau** contenant les attributs et les méthodes suivants :

- *lon* qui représente la longueur du bateau (nombre entier);
- *larg* qui représente la largeur du bateau (nombre entier);
- *v* qui représente la vitesse du bateau (nombre entier);
- *nbBateau* qui correspond au nombre des bateaux qui seront créés.
- Un constructeur permettant d'initialiser les attributs par des valeurs fournies en paramètres ;
- Un constructeur sans paramètres permettant d'initialiser les attributs à 0 en faisant appel au premier constructeur ;
- Un constructeur avec un paramètre *Bateau* permettant d'initialiser les attributs ;
- Une méthode *getNb()* qui retourne le nombre des objets créés de la classe *Bateau*.
- Une méthode *afficheBateau()* permettant d'afficher les caractéristiques de la classe *Bateau*.

**B.** Créer une classe **Sous-marin** qui dérive de la classe *Bateau*. Elle possède :

- L'attribut *prof* (nombre entier) qui représente la profondeur à laquelle il peut plonger le sous-marin.
- Un constructeur permettant d'initialiser les attributs de la classe par les valeurs fournies en paramètres, en utilisant le premier constructeur de la classe *Bateau*.
- La méthode *AfficheMarin()* permettant d'afficher les caractéristiques du sous-marin.
- Une méthode *compare()* pour la comparaison du sous-marin avec un autre sous-marin fournis en paramètre. Notons que 2 sous-marins sont égaux s'ils plongent à la même profondeur.

**C.** Implémenter une méthode *main* pour afficher les caractéristiques du bateau **B1**(140, 220, 120) et les caractéristiques des sous-marins **S1**(123, 230, 120, 200) et **S2**(150, 240, 120, 220), puis comparer les deux sous-marins S1 et S2 ; Ensuite, afficher le nombre des bateaux créés ;

## Partie Final (~ 105 min)

### Exercice 2 ( 55 pts )

Soit l'application web **Examen** contenant le WebForm « **WFPartiel.aspx** ».

Ecrire le script correspondant aux boutons « *ADD* », « *Edit* », « *Modify* », « *Delete Selected* » et « *Delete All* » :

WFPartiel.aspx\* X

### Examen Partiel

Nom :  Prenom :  Note (sur 100) :

Note doit etre entre 0 et 100 !

ADD

Liste des etudiants :

Unbound

Edit Modify Delete Selected Delete All

Le nb des etudiants : 0  
Le nb des etudiants reussis : 0

- Le bouton « ADD » (btnADD) permet d'ajouter l'étudiant dans la ListBox **lbEtud** (liste des étudiants) sous la forme de : *prenomX nomX a eu la note noteX*, puis vide les TextBox de nom (tbNom), prénom (tbPrenom) et note (tbNote). Par exemple : *Rola Zakaria a eu la note 67*
- Le bouton « Edit » (btnEdit) permet de recopier de la liste **lbEtud**, les informations, concernant l'étudiant sélectionné, dans les TextBox tbNom, tbPrenom et tbNote respectivement.
- Le bouton « Modify » (btnModify) permet de modifier les informations de l'étudiant sélectionné.
- Le bouton « Delete Selected » (btnDelSel) permet de supprimer de la liste, l'étudiant sélectionné.
- Le bouton « DeleteAll » (btnDelAll) permet de vider la liste des étudiants.
- Le nombre des étudiants est un label « lbINbEtud », initialisé à 0, et qui sera incrémenté (resp. décrémenté) à chaque ajout (resp. suppression) d'un étudiant dans (resp. de) la liste.
- Le nombre des étudiants réussis est un label « lbINbReussis », initialisé à 0, et qui sera incrémenté (resp. décrémenté) à chaque ajout (resp. suppression) d'un étudiant ayant une note  $\geq 50$  ou même de modification de la note.

### Exercice 3 ( 30 pts )

On ajoute à l'application Examen, le webForm « **WFResultat.aspx** ».

Nb des etudiants	Nb des Etudiants reussis	Nb des etudiants echoues
0	0	0
Pourcentage :	0	0

1. Comment Faire pour que la page « WFPartiel.aspx » soit la page de démarrage de l'application ?
2. Que signifie chacune des propriétés suivantes concernant le textbox *tbNom* , dans WFPartiel.aspx:  
`<asp:TextBox ID = "tbNom" runat = "server" MaxLength = "7" TextMode = "Password" ToolTip = "Nom"> </asp:TextBox>`
3. Quelle est la différence entre un évènement *Load* et un évènement *Init* d'une page ASP.net ?
4. Ecrire le script correspondant à l'évènement *Load* du webform WFResultat.aspx , afin de :
  - a) récupérer, de WFPartiel.aspx, le nombre des étudiants et l'afficher dans le label « *lblEtud* » ;
  - b) récupérer, de WFPartiel.aspx, le nombre des étudiants réussis (mettre dans le label *lblReussi*) ;
  - c) calculer le nombre des étudiants échoués et l'afficher dans le label « *lblEchoue* ».
  - d) calculer le pourcentage de réussite et l'afficher dans le label « *lblPR* ».
  - e) calculer le pourcentage d'échec et l'afficher dans le label « *lblPE* ».

### Exercice 4 ( 15 pts )

On ajoute à l'application Examen, le webForm « **WFBBase.aspx** ». Ecrire le script correspondant au bouton « *Ajouter* » (*btnAjouter*), permettant d'ajouter, dans la table *tbEtud*, les informations saisies par l'utilisateur. Les informations sont saisies dans les textbox *txtID*, *txtNom*, *txtAdresse* et *txtAge*, avec :

```
<asp:SqlDataSource ID="SqlDataSource1" runat="server"
    . . .
    ConnectionString="<%$ ConnectionStrings:ConnectionStringExam %>"
    . . .
    SelectCommand="SELECT * FROM [tbEtud] ORDER BY [ID]"
    . . .
</asp:SqlDataSource>
```

```

<asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
    DataKeyNames="ID" DataSourceID="SqlDataSource1">
    <Columns>
        <asp:BoundField DataField="ID" HeaderText="ID" InsertVisible="False"
            ReadOnly="True" SortExpression="ID" />
        <asp:BoundField DataField="Nom" HeaderText="Nom" SortExpression="Nom" />
        <asp:BoundField DataField="adresse" HeaderText="adresse"
            SortExpression="adresse" />
        <asp:BoundField DataField="age" HeaderText="age" SortExpression="age" />
        <asp:CommandField ShowEditButton="True" />
        <asp:CommandField ButtonType="Button" ShowDeleteButton="True" />
    </Columns>
</asp:GridView>

```

WFBase.aspx\* X

SqlDataSource - SqlDataSource1

**Liste des Etudiants :**

ID	Nom	adresse	age		
0	abc	abc	0	<a href="#">Edit</a>	Delete
1	abc	abc	1	<a href="#">Edit</a>	Delete
2	abc	abc	2	<a href="#">Edit</a>	Delete
3	abc	abc	3	<a href="#">Edit</a>	Delete
4	abc	abc	4	<a href="#">Edit</a>	Delete

ID Etudiant :  Nom :

Adresse :  Age :  **Age doit etre entre 0 et 59**

La liste des étudiants :

ID	Nom	adresse	age		
1	Ali	Tripoli	23	<a href="#">Edit</a>	Delete
2	Marwan	Beyrouth	34	<a href="#">Edit</a>	Delete
3	Joe	Saida	25	<a href="#">Edit</a>	Delete
4	Sami	Bekaa	32	<a href="#">Edit</a>	Delete
5	Maria	Tripoli	27	<a href="#">Edit</a>	Delete

**Bonne Chance**