

00186Fonaments d'Informàtica (103806) Curs 2020-

2021

Examen Parcial (23 de Novembre de 2020)

Grup:

Nom estudiant: JOSE ELSTON MARINÉ NIU: 4177900

Important: Recordeu que cal donar les millors solucions possibles en cada exercici. A més de funcionar correctament, el codi ha d'estar ben programat (codi clar, amb les instruccions més adients, sense operacions ni variables innecessàries, etc.) Les funcions i procediments de les preguntes de la 1 a la 6 formen part d'un únic programa i, per tant, ha d'haver coherentia entre les seves definicions, i la forma en què s'utilitzen en altres preguntes. El context del programa a desenvolupar, el tenui a continuació:

El Bingo de 75 boles és un joc d'atzar on hi ha 75 boles numerades de l'1 al 75 dins del bombo. Els cartrons tenen un total de 25 números. En aquest tipus de bingo, es van treient boles del bombo i es guanya un premi quan han sortit tots els números del nostre cartró. Això es coneix com a bingo o full House.

Volem fer un programa que permeti jugar al Bingo a dos jugadors. Els jugadors podran jugar tantes partides com vulguin i caldrà portar el compte de quantes partides ha guanyat cadascú.

Exercici 1 (1 punt)

Fer un procediment anomenat CrearCartró per a generar un cartró. S'han de generar 25 nombres del 1 al 75. S'ha d'assegurar que hi hagin:

- 5 nombres del 11 al 15,
- 5 nombres del 16 al 30,
- 5 nombres del 46 al 60 i
- 5 nombres del 61 al 75.

El procediment rebrà un array (que serà el cartró) i la dimensió de l'array. Retornarà l'array ple amb els números ordenats de menor a major.

UTILITZEU (no implementar) els següents procediments o funcions:

- int GenerarNombre (int min, int max, int all, int dim)
Genera un nombre aleatori entre min i max, i revisa que no estigui a l'array a de dimensió dim.
- void Ordena (int a[], int dim)

Ordena l'array a de dimensió dim de menor a major.

$$35 \times 15 = 525$$

$$75 + 25 = \underline{100}$$

Exercici 2 (1 punt)

Fer la funció anomenada ArraysIguals per saber si dos arrays tenen tots els valors idèntics. La funció rebrà dos arrays i la seva dimensió. Retornarà True en el cas que siguin iguals i False en cas contrari.

NOTA: Podeu suposar que els dos arrays tenen la mateixa dimensió.

$$325 + 150 = 475$$

$$\text{RESULTAT} : 35$$

Exercici 3 (1 punt)

Fer un procediment anomenat ImprimirBolesSortides per imprimir els nombres de les boles que ja han sortit. El procediment rebrà un array i la seva dimensió. L'array contindrà informació sobre si un nombre ha sortit o no. Si el valor d'una determinada posició (index) és un 1, voldrà dir que la bola corresponent ha sortit i si el és un 0, voldrà dir que aquella bola no ha sortit.

NOTA: Recordeu que els números de les boles van de 1 a 75.

Exercici 4 (1 punt)

Fer una funció anomenada Escrutini per saber si un cartró ha aconseguit Bingo. La funció rebrà un array que serà el cartró, la dimensió del cartró i un array (bombo) que contindrà informació dels noms que han sortit. En l'array bombo si el valor d'una determinada posició (index) és un 1, vindrà dir que la bola corresponent ha sortit i si el és un 0, vindrà dir que aquella bola no ha sortit.

La funció retornarà true en el cas que s'hagi aconseguit Bingo i false en cas contrari.

```
}
```

Mostra per pantalla un menú amb les opcions del codi: 1. Assignar cartrons, 2. Jugar, etc.

```
void InicialitzarArray(int a[], int dim, int v)
```

Inicialitza el vector a de dimensió dim al valor v

Retorna un enter aleatori entre min,max, controlant que no estigui a l'array a de dimensió dim

Exercici 5 (1 punt)

Fer un procediment anomenat ImprimirComentari per imprimir un comentari sobre el marcador. El procediment rebrà dos valors enters que seran el nombre de partides guanyades per cada jugador. En el cas que la diferència sigui:

- menor o igual que 1, escriure el missatge "Està molt igualat"
- entre 2 i 5, el missatge hauria de ser "La remuntada és possible".
- Major o igual que 6, el missatge haurà de ser "Avui és el teu dia de sort"

NOTA: podeu utilitzar la funció abs() per obtenir el valor absolut d'un nombre.

Exercici 5 (1 punt)

Fer un procediment anomenat ImprimirComentari per imprimir un comentari sobre el marcador. El procediment rebrà dos valors enters que seran el nombre de partides guanyades per cada jugador. En el cas que la diferència sigui:

- menor o igual que 1, escriure el missatge "Està molt igualat"
- entre 2 i 5, el missatge hauria de ser "La remuntada és possible".
- Major o igual que 6, el missatge haurà de ser "Avui és el teu dia de sort"

Exercici 6 (2.5 punts)

Suposeu que teniu els següents procediments i funcions ja implementats:

```
void Menuprincipal()
{
    cout << "---- Menu Principal ----" << endl;
    cout << "1.- Asignar cartrons " << endl;
    cout << "2.- Jugar" << endl;
    cout << "3.- Marcador" << endl;
    cout << "4.- Sortir" << endl;
}
```

Exercici 7 (2.5 punts)

Fer un programa complet (declaracions globals i funció main()) que segueixi els següents passos:

- Declarar (on corresponguï) les constants necessàries per tal que el programa sigui fàcilment modificable.
- Declarar (on corresponguï) els arrays necessaris per representar els cartrons de dos jugadors anomenats cartróPlayer1 i cartróPlayer2. Declarar també un array, anomenat bombo, per controlar quins nombres han sortit i quins no.
- Utilitzar el procediment Menuprincipal() i llegir l'opció escollida.
- Si l'opció és 1, generar dos cartrons (un per a cada jugador) seguint els passos següents:
 - Initialitzar els cartrons dels jugadors tot a 0 amb el procediment InicialitzarArray.
 - Utilitzar el procediment CrearCartró per crear els cartrons del jugador 1 i del jugador 2.
 - Comprovar si els dos cartrons són diferents utilitzant la funció ArraysIquals. En cas que siguin iguals tornar a generar un dels dos.
- Si l'opció és 2, implementa el joc seguint els passos següents:
 - S'haurà de comprovar que s'han generat els cartrons (s'ha entrat a l'opció 1). En cas contrari, donar un missatge d'error i tornar al menú principal sense fer res més.
 - Inicialitzar l'array bombo a 0 amb el procediment InicialitzarArray.
 - Generar un nombre per simular la bola amb la funció GenerarNombre.
 - Imprimir el nombre que ha sortit i utilitzar la funció ImprimirBolesSortides per imprimir els noms de les bales que han sortit al llarg de la partida.
 - Utilitzar la funció Escrutini per comprovar si algun dels dos jugadors ha aconseguit fer Bingo.
 - En cas que no hi hagi cap guanyador tornar al punt 5.3
 - Si l'opció és 3: imprimir el marcador de les partides que s'han jugat i mostrar el missatge generat pel procediment ImprimirComentari.
 - Si l'opció és 4, imprimir el marcador final i mostrar el missatge: "Sortint del programa..."
 - Qualsevol altra opció, escriure el missatge: "Opció no permesa".
 - Repetir els passos 3 a 9, fins que l'opció del menú escollida sigui la 4.

Exercici 7 (0.5 punts)

Declarar un nou tipus de dades, `TCartro`, com un registre amb els camps: `Ident_Cartro` (valor enter), `Numeros` (array de 25 enters) i `Nom_Jugador` (cadena de 50 caràcters). Fer un procediment anomenat `Imprimir` que rebi com a paràmetre un registre del tipus `TCartro` i imprimeixi l'identificador del cartró i el nom del jugador separant -los amb el caràcter '-'.

Exercici 9 (1 punt)

Escrivre la funció `Elevar` que retorna el resultat d'elevat un nombre al quadrat. La funció tindrà dos paràmetres de tipus int. El primer serà la base i el segon serà el resultat de l'operació en el cas que es pugui realitzar. La funció retornarà un 1 si s'ha pogut realitzar l'operació i un 0 en cas contrari.

NOTA: El valor de la base ha d'estar inclos en l'interval [-46,340,46,340] per evitar overflow.

Exercici 8 (1 punt)

Fer una funció anomenada `Es_MatPositiva` que rebi com a paràmetres: una matriu d'enters de 4 columnes, el número de files de la matriu, i el número de columnes de la matriu. i retorna 1 si tots els valors de la matriu són positius o zero, i un 0 en cas contrari.

Exercicis d'informàtica (103806)

Curs 2020-2021

Examen Parcial (23 de Novembre de 2020)

Grup: 5 3 6 4 4

Nom estudiant: Miquel Pau Solerons

Exercici 2 (1 punt)
Fer la funció anomenada `ArraysIguals` per saber si dos arrays tenen tots els valors idèntics.
La funció rebrà dos arrays i la seva dimensió. Retornarà `True` en el cas que siguin iguals i `False` en cas contrari.

NOTA: Podeu suposar que els dos arrays tenen la mateixa dimensió.

Important: Recordeu que cal donar les millors solucions possibles en cada exercici. A més de funcionar correctament, el codi ha d'estar ben programat (codi clar, amb les instruccions més adients, sense operacions ni variables innecessàries, etc.)

Les funcions i procediments de les preguntes de la 1 a la 6 formen part d'un únic programa i, per tant, ha d'haver coherentia entre les seves definicions, i la forma en què s'utilitzen en altres preguntes. El context del programa a desenvolupar, el teniu a continuació:

El Bingo de 75 boles és un joc d'atzar on hi ha 75 boles numerades de l'1 al 75 dins del bumbo.

Els cartells tenen un total de 25 nombres. En aquest tipus de bingo, es van treient boles del bumbo i es guanya un premi quan han sortit tots els nombres del nostre cartell. Això es coneix com a bingo o full House.

Volem fer un programa que permeti jugar al Bingo a dos jugadors. Els jugadors podran jugar tantes partides com vulguin i caldrà portar el compte de quantes partides ha guanyat cadascú.

Exercici 1 (1 punt)

Fer un procediment anomenat `CrearCartell` per a generar un cartell. S'han de generar 25 nombres del 1 al 75. S'ha d'assegurar que hi hagin:

- 5 nombres del 11 al 15,
- 5 nombres del 46 al 60 i
- 5 nombres del 61 al 75.
- 5 nombres del 31 al 45,

El procediment rebrà un array (que serà el cartell) i la dimensió de l'array. Retornarà l'array ple amb els números ordenats de menor a major.

UTILITZEU (no implementar) els següents procediments o funcions:

- `int GenerarNombre (int min, int max, int a[], int dim)`
Genera un nombre aleatori entre min,max, revisa que no estigui a l'array a de dimensió dim.
- `void Ordena (int a[], int dim)`
Ordena l'array a de dimensió dim de menor a major.

$$15 + 15 = \underline{30}$$

$$35 - 10 = \underline{15}$$

Exercici 4 (1 punt)

Fer una funció anomenada Escrutini per saber si un cartró ha aconseguit Bingo. La funció rebrà un array que serà el cartró, la dimensió del cartró i un array (bombo) que contindrà informació dels nombres que han sortit. En l'array bombo si el valor d'una determinada posició (index) és un 1, vindrà dir que la bola corresponent ha sortit si el és un 0, vindrà dir que aquella bola no ha sortit.

La funció retornarà true en el cas que s'hagi aconseguit Bingo i false en cas contrari.

```
}
```

Exercici 6 (2.5 punts)

Suposeu que teniu els següents procediments i funcions ja implementats:

```
void MenúPrincipal()
{
    cout << " --- Menú Principal --- " << endl;
    cout << "1.- Asignar cartrons " << endl;
    cout << "2.- Jugar " << endl;
    cout << "3.- Marcador" << endl;
    cout << "4.- Sortir" << endl;
}
```

Mostra per pantalla un menú amb les opcions del codi: 1. Asignar cartrons, 2. Jugar, etc.

```
void InicialitzarArray(int a[], int dim, int v)
```

Inicialitza el vector a de dimensió dim al valor v

```
int GenerarNombre(int min, int max, int a[], int dim)
Retorna un enter aleatori entre min i max, controlant que no estigui a l'array a de dimensió dim
```

Fer un programa complet (declaracions globals i funció main()) que segueixi els següents passos:

1. Declarar (on correspongui) les constants necessàries per tal que el programa sigui fàcilment modificable.
2. Declarar (on correspongui) els arrays necessaris per representar els cartrons de dos jugadors anomenats cartroPlayer1 i cartroPlayer2. Declarar també un array, anomenat bombo, per controlar quins nombres han sortit i quins no.
3. Utilitzar el procediment MenúPrincipal() i llegir l'opció escollida.
4. Si l'opció és 1, generar dos cartrons (un per a cada jugador) seguint els passos següents:
 - 4.1. Inicialitzar els cartrons dels jugadors tot a 0 amb el procediment InicialitzarArray.
 - 4.2. Utilitzar el procediment CrearCartro per crear els cartrons del jugador 1 i del jugador 2.
 - 4.3. Comprovar si els dos cartrons són diferents utilitzant la funció ArraysIguals. En cas que siguin iguals tornar a generar un dels dos.
5. Si l'opció és 2, implementa el joc seguint els passos següents:
 - 5.1. S'haurà de comprovar que s'han generat els cartrons (s'ha entrat a l'opció 1). En cas contrari, donar un missatge d'error i tornar al menú principal sense fer res més.
 - 5.2. Inicialitzar l'array bombo a 0 amb el procediment InicialitzarArray.
 - 5.3. Generar un nombre per simular la bola amb la funció GenerarNombre.
 - 5.4. Imprimir el nombre que ha sortit i utilitzar la funció ImprimirBolesSortides per imprimir els nombres de les boles que han sortit al llarg de la partida.
 - 5.5. Utilitzar la funció Escrutini per comprovar si algun dels dos jugadors ha aconseguit fer Bingo.
 - 5.6. En cas que no hi hagi cap guanyador tornar al punt 5.3
6. Si l'opció és 3: imprimir el marcador de les partides que s'han jugat i mostrar el missatge generat pel procediment ImprimirComentari.
7. Si l'opció és 4, imprimir el marcador final i mostrar el missatge: "Sortint del programa...".
8. Qualsevol altra opció, escriure el missatge: "Opció no permesa".
9. Repetir els passos 3 a 9, fins que l'opció del menú escollida sigui la 4.

Exercici 7 (0.5 punts)

Declarar un nou tipus de dades, `TCartró`, com un registre amb els camps: `Ident_Cartró` (valor enter), `Numeros` (array de 25 enters) i `Nom_Jugador` (cadena de 50 caràcters). Fer un procediment anomenat `Imprimir` que rebi com a paràmetre un registre del tipus `TCartró` i imprimeixi l'identificador del cartró i el nom del jugador separant-los amb el caràcter ‘,’.

Exercici 8 (1 punt)

Fer una funció anomenada `Es_MatPositiu` que rebi com a paràmetres: una matriu d'enters de 4 columnes, el número de files de la matriu, i el número de columnes de la matriu, i retorna 1 si tots els valors de la matriu són positius o zero, i un 0 en cas contrari.

Exercici 9 (1 punt)

Escriure la funció `Elevar` que retorna el resultat d'elevat un nombre al quadrat. La funció tindrà 2 paràmetres de tipus int. El primer serà la base i el segon serà el resultat de l'operació en el cas que es pugui realitzar. La funció retornarà un 1 si s'ha pogut realitzar l'operació i un 0 en cas contrari.

NOTA: El valor de la base ha d'estar incòs en l'intervall [-46.340,46.340] per evitar overflow.

J0186Fonaments d'Informàtica (103806) Curs 2020-

2021

Examen Parcial (23 de Novembre de 2020)

Grup: NIU: 3454743

Nom estudiant: DOMÈNEC VÉTELL

Exercici 2 (1 punt)
Fer la funció anomenada `ArraysIguals` per saber si dos arrays tenen tots els valors idèntics. La funció rebrà dos arrays i la seva dimensió. Retornarà `True` en el cas que siguin iguals i `False` en cas contrari.

NOTA: Podeu suposar que els dos arrays tenen la mateixa dimensió.

Important: Recordeu que cal donar les millors solucions possibles en cada exercici. A més de funcionar correctament, el codi ha d'estar ben programat (codi clar, amb les instruccions més adients, sense operacions ni variables innecessàries, etc.) Les funcions i procediments de les preguntes de la 1 a la 6 formen part d'un únic programa i, per tant, ha d'haver coherència entre les seves definicions, i la forma en què s'utilitzen en altres preguntes. El context del programa a desenvolupar, el teniu a continuació:

El Bingo de 75 boles és un joc d'atzar on hi ha 75 boles numerades de l'1 al 75 dins del bombo. Els cartíons tenen un total de 25 números. En aquest tipus de bingo, es van treient boles del bombo i es guanya un premi quan han sortit tots els números del nostre cartíol. Això es coneix com a bingo o full House.

Volem fer un programa que permeti jugar al Bingo a dos jugadors. Els jugadors podran jugar tantes partides com vulguin i caldrà portar el compte de quantes partides ha guanyat cadascú.

Exercici 1 (1 punt)

Fer un procediment anomenat `CrearCartó` per a generar un cartíol. S'han de generar 25 nombres del 1 al 75. S'ha d'assegurar que hi hagin:

- 5 nombres del 11 al 15,
- 5 nombres del 46 al 60 i
- 5 nombres del 61 al 75.

El procediment rebrà un array (que serà el cartíol) i la dimensió de l'array. Retornarà l'array ple amb els números ordenats de menor a major.

UTILITZEU (no implementar) els següents procediments o funcions:

- `int GeneraNombre (int min, int max, int a[], int dim)`
Genera un nombre aleatori entre min i max, i revisa que no estigui a l'array a de dimensió dim.
- `void Ordena (int a[], int dim)`
Ordena l'array a de dimensió dim de menor a major.

$$25 \times 25 = 2125$$

$$100 - 35 = \boxed{65}$$

$$10 + 10 = 20$$

$$2 \times 10 = 20$$

$$2 \times 10 = 20$$

Exercici 3 (1 punt)

Fer un procediment anomenat `ImprimirBolesSortides` per imprimir els nombres de les boles que ja han sortit. El procediment rebrà un array i la seva dimensió. L'array contindrà informació sobre si un nombre ha sortit o no. Si el valor d'una determinada posició (`index`) és un 1, voldrà dir que la bola corresponent ha sortit; si el és un 0, voldrà dir que aquella bola no ha sortit.

NOTA: Recordeu que els números de les boles van de 1 a 75.

Exercici 4 (1 punt)
 Fer una funció anomenada Escrutini per saber si un cartó ha aconseguit Bingo. La funció rebrà un array que serà el cartó, la dimensió del cartó i un array (bombo) que contindrà informació dels nombres que han sortit. En l'array bombo si el valor d'una determinada posició (index) és un 1, voldrà dir que la bola corresponent ha sortit i si el és un 0, voldrà dir que aquella bola no ha sortit.

La funció retornarà true en el cas que s'hagi aconseguit Bingo i false en cas contrari.

Exercici 6 (2,5 punts)
 Suposeu que tenui els següents procediments i funcions ja implementats:

```
void InicialitzarArray(int min, int max, int a[], int dim)
{
    cout << "---- Menú Principal ----" << endl;
    cout << "1.- Asignar cartons" << endl;
    cout << "2.- Jugar" << endl;
    cout << "3.- Marcador" << endl;
    cout << "4.- Sortir" << endl;
}
```

Mostra per pantalla un menú amb les opcions del codi: 1. Assignar cartons, 2. Jugar, etc.

```
void InicialitzarPrincipal()
{
    cout << "---- Menú Principal ----" << endl;
    cout << "1.- Asignar cartons" << endl;
    cout << "2.- Jugar" << endl;
    cout << "3.- Marcador" << endl;
    cout << "4.- Sortir" << endl;
}
```

voide InicialitzarPrincipal()

int GenerarNombre (int min, int max, int a[], int dim)
 Retorna un enter aleatori entre min i max, controlant que no estigui a l'array a de dimensió dim

Inicializa el vector a de dimensió dim al valor v

int main()
{
 int min, max, a[10], dim;

Fer un programa complet (declaracions globals i funció main()) que segueixi els següents passos:

- Declarar (on corresponguí) les constants necessàries per tal que el programa sigui fàcilment modificable.
- Declarar (on corresponguí) els arrays necessaris per representar els cartons de dos jugadors anomenats cartoPlayer1 i cartoPlayer2. Declarar també un array, anomenat bombo, per controlar quins nombres han sortit i quins no.
- Utilitzar el procediment MenuPrincipal () i llegir l'opció escollida.
- Si l'opció és 1, generar dos cartons (un per a cada jugador) seguint els passos següents:
 - Inicialitzar els cartons dels jugadors tot a 0 amb el procediment InicialitzarArray.
 - Utilitzar el procediment CreateCarto per crear els cartons del jugador 1 i del jugador 2.
 - Comprovar si els dos cartons són diferents utilitzant la funció ArraysIguals. En cas que siguin iguals tornar a generar un dels dos.
 - Si l'opció és 2, implementa el joc seguint els passos següents:
 - Si l'opció és 2, comprovar que s'han generat els cartons (s'ha entrat a l'opció 1). En cas contrari, donar un missatge d'error i tornar al menú principal sense fer res més.
 - Inicialitzar l'array bombo a 0 amb el procediment InicialitzarArray.
 - Generar un nombre per simular la bola amb la funció GenerarNombre.
 - Imprimir el nombre que ha sortit i utilitzar la funció ImprimirBolesSortides per imprimir els noms de les boles que han sortit al llarg de la partida.
 - Utilitzar la funció Escrutini per comprovar si algun dels dos jugadors ha aconseguit fer Bingo.
 - En cas que no hi hagi cap guanyador tornar al punt 5.3
 - Si l'opció és 3: imprimir el marcador de les partides que s'han jugat i mostrar el missatge generat pel procediment ImprimirComentari.
 - Si l'opció és 4, imprimir el marcador final i mostrar el missatge: "Sortint del programa...".
 - Qualsevol altra opció, escriure el missatge: "Opció no permetuda".
 - Repetir els passos 3 a 9, fins que l'opció del menú escollida sigui la 4.

Exercici 7 (0.5 punts)

Declarar un nou tipus de dades, `TCarro`, com un registre amb els camps: `Ident_Carro` (valor enter), `Numeros` (array de 25 enters) i `Nom_Jugador` (cadena de 50 caràcters). Fer un procediment anomenat `Imprimir` que rebi com a paràmetre un registre del tipus `TCarro` i imprimeixi l'identificador del carro i el nom del jugador separant-los amb el caràcter '-'.

Exercici 8 (1 punt)

Exercici 8 (1 punt)
Fer una funció anomenada `E_s_MatPosit` va que rebi com a paràmetres: una matriu d'enters de 4 columnes, el número de files de la matriu i el número de columnes de la matriu, i retorna 1 si tots els valors de la matriu són positius o zero, i un 0 en cas contrari.

Exercici 9 (1 punt)

Escrivire la funció `Elevar` que retorni el resultat d'elevat un nombre al quadrat. La funció tindrà dos paràmetres de tipus int. El primer serà la base i el segon serà el resultat de l'operació en el cas que es pugui realitzar. La funció retornara un 1 si s'ha pogut realitzar l'operació i un 0 en cas contrari.

NOTA: El valor de la base ha d'estar inclos en l'interval [-46.340, 46.340] per evitar overflow.

Onaments d'Informàtica (103806)

Examen Parcial (23 de Novembre de 2020)

Curs 2020-2021

Nom estudiant: Jones Davis Trudeau Grup: 5089481 NIU: 5089481

Important: Recordeu que cal donar les millors solucions possibles en cada exercici. A més de funcionar correctament, el codi ha d'estar ben programat (codi clar, amb les instruccions més adients, sense operacions ni variables innecessàries, etc.)

Les funcions i procediments de les preguntes de la 1 a la 6 formen part d'un únic programa i, per tant, ha d'haver coherència entre les seves definicions, i la forma en què s'utilitzen en altres preguntes. El context del programa a desenvolupar, el teniu a continuació:

El Bingo de 75 boles és un joc d'aixar on hi ha 75 boles numerades de l'1 al 75 dins del bombo. Els cartells tenen un total de 25 nombres. En aquest tipus de bingo, es van treient boles del bombo i es guanya un premi quan han sortit tots els nombres del nostre cartell. Això es coneix com a bingo o full House.

Volem fer un programa que permeti jugar al Bingo a dos jugadors. Els jugadors podran jugar tantes partides com vulguin i caldrà portar el compte de quantes partides ha guanyat cadascú.

Exercici 1 (1 punt)

Fer un procediment anomenat `CrearCartell` per a generar un cartell. S'han de generar 25 nombres del 1 al 75. S'ha d'assegurar que hi hagin:

- 5 nombres del 1 al 15.
- 5 nombres del 16 al 30.
- 5 nombres del 31 al 45.
- 5 nombres del 46 al 60 i
- 5 nombres del 61 al 75.

El procediment rebrà un array (que serà el cartell) i la dimensió de l'array. Retornarà l'array ple amb els números ordenats de menor a major.

UTILITZEU (no implementar) els següents procediments o funcions:

- `int GenerarNombre (int min, int max, int a[], int dim)`
Genera un nombre aleatori entre min i max, i revisa que no estigui a l'array a de dimensió dim.
- `void Ordena (int a[], int dim)`
Ordena l'array a de dimensió dim de menor a major.

$$4 \times 10 = \underline{\underline{40}}$$
$$40 - 9 = \underline{\underline{31}}$$

Fer la funció anomenada `ArraysIguals` per saber si dos arrays tenen tots els valors idèntics. La funció rebrà dos arrays i la seva dimensió. Retornarà `true` en el cas que siguin iguals i `false` en cas contrari.

NOTA: Podeu suposar que els dos arrays tenen la mateixa dimensió.

$$445 - 20 = 425$$

$$\text{Resultat: } 100$$

Exercici 3 (1 punt)

Fer un procediment anomenat `ImprimirBolesSortides` per imprimir els nombres de les boles que ja han sortit. El procediment rebrà un array i la seva dimensió. L'array contingrà informació sobre si un nombre ha sortit o no. Si el valor d'una determinada posició (index) és un 1, vindrà dir que la bola corresponent ha sortit i si el és un 0, vindrà dir que aquella bola no ha sortit.

NOTA: Recordeu que els números de les boles van de 1 a 75.

Exercici 4 (1 punt)
Fer una funció anomenada Escrutini per saber si un cartró ha aconseguit Bingo. La funció rebrà un array que serà el cartró, la dimensió del cartró i un array (bombo) que contindrà informació dels nombres que han sortit. En l'array bombó si el valor d'una determinada posició (index) és un 1, voldrà dir que la bola corresponent ha sortit i si el és un 0, voldrà dir que aquella bola no ha sortit.

La funció retornarà true en el cas que s'hagi aconseguit Bingo i False en cas contrari.

Exercici 6 (2 punts)
Suposeu que tenu els següents procediments i funcions ja implementats:

```
void MenuPrincipal()
{
    cout << "---- Menú Principal ----" << endl;
    cout << "1.- Asignar cartrons" << endl;
    cout << "2.- Jugar" << endl;
    cout << "3.- Marcador" << endl;
    cout << "4.- Sortir" << endl;
}
```

Mostra per pantalla un menú amb les opcions del codi: 1. Assignar cartrons, 2. Jugar, etc.

```
void InicialitzarArray(int a[], int dim, int v)
{
    int GenerarNombre (int min, int max, int a[], int dim)
```

Retorna un enter aleatori entre min i max, controlant que no estigui a l'array a de dimensió dim

Fer un programa complet (declaracions globals i funció main()) que segueixi els següents passos:

1. Declarar (on correspongui) les constants necessàries per tal que el programa sigui fàcilment modificable.
2. Declarar (on correspongui) els arrays necessaris per representar els cartrons de dos jugadors anomenats cartronPlayer1 i cartronPlayer2. Declarar també un array, anomenat bombo, per controlar quins nombres han sortit i quins no.
3. Utilitzar el procediment MenuPrincipal() i llegir l'opció escollida.
4. Si l'opció és 1, generar dos cartrons (un per a cada jugador) seguint els passos següents:
 - 4.1. Inicialitzar els cartrons dels jugadors tot a 0 amb el procediment InicialitzarArray.
 - 4.2. Utilitzar el procediment CrearCartró per crear els cartrons del jugador 1 i del jugador 2.
 - 4.3. Comprovar si els dos cartrons són diferents utilitzant la funció ArraysIguals. En cas que siguin iguals tornar a generar un dels dos.
5. Si l'opció és 2, implementa el joc seguint els passos següents:
 - 5.1. S'haurà de comprovar que s'han generat els cartrons (s'ha entrat a l'opció 1). En cas contrari, donar un missatge d'error i tornar al menú principal sense fer res més.
 - 5.2. Inicialitzar l'array bombo a 0 amb el procediment InicialitzarArray.
 - 5.3. Generar un nombre per simular la bola amb la funció GenerarNombre.
 - 5.4. Imprimir el nombre que ha sortit i utilitzar la funció ImprimirBolesSortides per imprimir els nombres de les boles que han sortit al llarg de la partida.
 - 5.5. Utilizar la funció Escrutini per comprovar si algun dels dos jugadors ha aconseguit fer Bingo.
 - 5.6. En cas que no hi hagi cap guanyador tornar al punt 5.3
 6. Si l'opció és 3: imprimir el marcador de les partides que s'han jugat i mostrar el missatge generat pel procediment ImprimirComentari.
 7. Si l'opció és 4, imprimir el marcador final i mostrar el missatge: "Sortint del programa..."
 8. Qualsevol altra opció, escriure el missatge: "Opció no permetuda".
 9. Repetir els passos 3 a 9, fins que l'opció del menú escollida sigui la 4.

Exercici 7 (0.5 punts)

Declarar un nou tipus de dades, `TCartro`, com un registre amb els camps: `Ident_Cartro` (valor enter), `Numeros` (array de 25 enters) i `Nom_Jugador` (cadena de 50 caràcters). Fer un procediment anomenat `Imprimir` que rebi com a paràmetre un registre del tipus `TCartro` i imprimeixi l'identificador del cartró i el nom del jugador separant-los amb el caràcter '-'.

Exercici 9 (1 punt)

Escriure la funció `Elevar` que retorni el resultat d'elevat un nombre al quadrat. La funció tindrà dos paràmetres de tipus int. El primer serà la base i el segon serà el resultat de l'operació en el cas que es pugui realizar. La funció retornara un 1 si sha pogut realizar l'operació i un 0 en cas contrari.

NOTA: El valor de la base ha d'estar inclos en l'interval [-46.340,46.340] per evitar overflow.

Exercici 8 (1 punt)

Fer una funció anomenada `Es_MatPositiva` que rebi com a paràmetres: una matriu d'enters de 4 columnes, el número de files de la matriu, i el número de columnes de la matriu, i retorna 1 si tots els valors de la matriu són positius o zero, i un 0 en cas contrari.

Examen Parcial (23 de Novembre de 2020)

Nom estudiant: FRANCINE FLORES Grup:

NIU: 1887624

Exercici 2 (1 punt)
 Fer la funció anomenada `ArraysIguals` per saber si dos arrays tenen tots els valors idèntics.
 La funció rebrà dos arrays i la seva dimensió. Retornarà `True` en el cas que siguin iguals i `False` en cas contrari.

NOTA: Podeu suposar que els dos arrays tenen la mateixa dimensió.

Important: Recordeu que cal donar les millors solucions possibles en cada exercici. A més de funcionar correctament, el codi ha d'estar ben programat (codi clar, amb les instruccions més adients, sense operacions ni variables innecessàries, etc.)
 Les funcions i procediments de les preguntes de la 1 a la 6 formen part d'un únic programa i, per tant, ha d'haver coherentia entre les seves definicions, i la forma en què s'utilitzen en altres preguntes. El context del programa a desenvolupar, el teniu a continuació:

El Bingo de 75 boles és un joc d'atzar on hi ha 75 boles numerades de l'1 al 75 dins del bombo. Els cartrons tenen un total de 25 nombres. En aquest tipus de bingo, es van treient boles del bombo i es guanya un premi quan han sortit tots els nombres del nostre cartó. Això es coneix com a bingo o full House.

Volem fer un programa que permeti jugar al Bingo a dos jugadors. Els jugadors podran jugar quantes partides com vulguin i caldrà portar el compte de quantes partides ha guanyat cadascú.

$$30 + 40 = 70$$

RESULTAT = 70

Exercici 1 (1 punt)

Fer un procediment anomenat `CrearCartó` per a generar un cartó. S'han de generar 25 nombres del 1 al 75. S'ha d'assegurar que hi hagin:

- 5 nombres del 1º al 15,
- 5 nombres del 16º al 30,
- 5 nombres del 31º al 45,
- 5 nombres del 46º al 60 i
- 5 nombres del 61º al 75.

El procediment rebrà un array (que serà el cartó) i la dimensió de l'array. Retornarà l'array ple amb els números ordenats de menor a major.

UTILITZEU (no implementar) els següents procediments o funcions:

- `int GenerarNombre(int min, int max, int a[], int dim)`
 Genera un nombre aleatori entre min i max, i revisa que no estigui a l'array a de dimensió dim.
- `void Ordena(int a[], int dim)`
 Ordene l'array a de dimensió dim de menor a major.

$$10 + 10 = 20$$

35 + 10 = 45

Exercici 3 (1 punt)

Fer un procediment anomenat `ImprimirBolesSortides` per imprimir els nombres de les boles que ja han sortit. El procediment rebrà un array i la seva dimensió. L'array contindrà informació sobre si un nombre ha sortit o no. Si el valor d'una determinada posició (`index`) és un 1, voldrà dir que la bola corresponent ha sortit; si el és un 0, voldrà dir que aquella bola no ha sortit.

NOTA: Recordeu que els números de les boles van de 1 a 75.

Exercici 4 (1 punt)

Fer una funció anomenada `Escrutini` per saber si un cartó ha aconseguit Bingo. La funció rebrà un array que serà el cartó, la dimensió del cartó i un array (`bombo`) que contendrà informació dels nombres que han sortit. En l'array `bombo` si el valor d'una determinada posició (`index`) és un 1, vindrà dir que la bola corresponent ha sortit i si el és un 0, vindrà dir que aquella bola no ha sortit.

La funció retornarà `True` en el cas que s'hagi aconseguit Bingo i `False` en cas contrari.

Exercici 6 (2.5 punts)

Suposeu que teniu els següents procediments i funcions ja implementats:

```
void MenuPrincipal()
{
    cout << " --- Menu Principal --- " << endl;
    cout << " 1. - Asignar cartons " << endl;
    cout << " 2. - Jugar" << endl;
    cout << " 3. - Marcador" << endl;
    cout << " 4. - Sortir" << endl;
}
```

Mostra per pantalla un menú amb les opcions del codi: 1. Assignar cartons, 2. Jugar, etc.

```
void InicialitzarArray(int a[], int dim, int v)
```

Retorna un enter aleatori entre `min` i `max`, controlant que no estigui a l'array a de dimensió `dim`. Inicialitza el vector a de dimensió `dim`.

Fer un programa complet (declaracions globals i funció `main()`) que segueixi els següents passos:

1. Declarar (on corresponguï) les constants necessàries per tal que el programa sigui fàcilment modificable.
2. Declarar (on corresponguï) els arrays necessaris per representar els cartons de dos jugadors anomenats `cartroPlayer1` i `cartroPlayer2`. Declarar també un array, anomenat `bombo`, per controlar quins nombres han sortit i quins no.
3. Utilitzar el procediment `MenuPrincipal()` i llegir l'opció escollida.
4. Si l'opció és 1, generar dos cartons (un per a cada jugador) seguint els passos següents:
 - 4.1. Inicialitzar els cartons dels jugadors tot a 0 amb el procediment `InicialitzarArray`.
 - 4.2. Utilitzar el procediment `CrearCarto` per crear els cartons del jugador 1 i del jugador 2.
 - 4.3. Comprovar si els dos cartons són diferents utilitzant la funció `ArraysIguals`. En cas que siguin iguals tornar a generar un dels dos.
5. Si l'opció és 2, implementa el joc seguint els passos següents:
 - 5.1. S'haurà de comprovar que s'han generat els cartons (s'ha entrat a l'opció 1). En cas contrari, donar un missatge d'error i tornar al menú principal sense fer res més.
 - 5.2. Inicialitzar l'array `bombo` a 0 amb el procediment `InicialitzarArray`.
 - 5.3. Generar un nombre per simular la bola amb la funció `GenerarNombre`.
 - 5.4. Imprimir el nombre que ha sortit i utilitzar la funció `ImprimirBolesSortides` per imprimir els nombres de les boles que han sortit al llarg de la partida.
 - 5.5. Utilitzar la funció `Escrutini` per comprovar si algun dels dos jugadors ha aconseguit fer Bingo.
 - 5.6. En cas que no hi hagi cap guanyador tornar al punt 5.3
6. Si l'opció és 3: imprimir el marcador de les partides que s'han jugat i mostrar el missatge generat pel procediment `ImprimirComentari`.
7. Si l'opció és 4, imprimir el marcador final i mostrar el missatge: "Sortint del programa...".
8. Qualsevol altra opció, escriure el missatge: "Opció no permesa".
9. Repetir els passos 3 a 9, fins que l'opció del menú escollida sigui la 4.

Exercici 7 (0.5 punts)

Declarar un nou tipus de dades, `TCartro`, com un registre amb els camps: `Ident_Cartro` (valor enter), `Numeros` (array de 25 enters) i `Nom_Jugador` (cadena de 50 caràcters).
Fer un procediment anomenat `Imprimir` que rebi com a paràmetre un registre del tipus `TCartro` i imprimeix l'identificador del cartó i el nom del jugador separant-los amb el caràcter ‘.

Exercici 9 (1 punt)

Escrivre la funció `Elevar` que retorna el resultat d'elevat un nombre al quadrat. La funció tindrà dos paràmetres de tipus `int`. El primer serà la base i el segon serà el resultat de l'operació en el cas que es pugui realizar. La funció retornarà un 1 si s'ha pogut realitzar l'operació i un 0 en cas contrari.

NOTA: El valor de la base ha d'estar inclòs en l'intervall [-46,340,46,340] per evitar overflow.

Exercici 8 (1 punt)

Fer una funció anomenada `E_S_MatPositiva` que rebi com a paràmetres: una matriu d'enters de 4 columnes, el número de files de la matriu, i el número de columnes de la matriu, i retorna 1 si tots els valors de la matriu són positius o zero, i un 0 en cas contrari.

Exercicis d'Informàtica (103806)

Curs 2020-2021

Examen Parcial (23 de Novembre de 2020)

Grup:

Nom estudiant: RODRIGO BROCÉN HUERTA NIU: 8163427

Important: Recordeu que cal donar les millors solucions possibles en cada exercici. A més de funcionar correctament, el codi ha d'estar ben programat (codi clar, amb les instruccions més adients, sense operacions ni variables innecessàries, etc.)

Les funcions i procediments de les preguntes de la 1 a la 6 formen part d'un únic programa i, per tant, ha d'haver coherentia entre les seves definicions, i la forma en què s'utilitzen en altres preguntes. El context del programa a desenvolupar, el teniu a continuació:

El Bingo de 75 boles és un joc d'atzar on hi ha 75 boles numerades de l'1 al 75 dins del baró. Els cartrons tenen un total de 25 nombres. En aquest tipus de bingo, es van treient boles del baró i es guanya un premi quan han sortit tots els nombres del nostre cartó. Això es coneix com a bingo o full House.

Volem fer un programa que permeti jugar al Bingo a dos jugadors. Els jugadors podran jugar tantes parties com vulguin i caldrà portar el compte de quantes parties ha guanyat cadascú.

Exercici 1 (1 punt)

Fer un procediment anomenat `CrearCartó` per a generar un cartó. S'han de generar 25 nombres del 1 al 75. S'ha d'assegurar que hi hagin:

- 5 nombres del 1 al 15,
- 5 nombres del 16 al 30,
- 5 nombres del 31 al 45,
- 5 nombres del 46 al 60 i
- 5 nombres del 61 al 75.

El procediment rebrà un array (que serà el cartó) i la dimensió de l'array. Retornarà l'array ple amb els números ordenats de menor a major.

UTILITZEU (no implementar) els següents procediments o funcions:

- `int GenerarNombre (int min, int max, int a[], int dim)`
Genera un nombre aleatori entre min i max, i revisa que no estigui a l'array a de dimensió dim.
- `void Ordena (int a[], int dim)`
Ordena l'array a de dimensió dim de menor a major.

$$31 + 31 = 62$$

$$90 + 10 = \boxed{100}$$

$$98 + 3 = 101$$

$$5 \times 5 = 25$$

$$\text{RESULTAT} = 126$$

Exercici 3 (1 punt)

Fer un procediment anomenat `ImprimirBolesSortides` per imprimir els nombres de les boles que ja han sortit. El procediment rebrà un array i la seva dimensió. L'array contingrà informació sobre si un nombre ha sortit o no. Si el valor d'una determinada posició (`Index`) és un 1, voldrà dir que la bola corresponent ha sortit; si el és un 0, voldrà dir que aquella bola no ha sortit.

NOTA. Recordeu que els números de les boles van de 1 a 75.

L'exercici 4 (1 punt)
Fer la funció anomenada `ArraysIguals` per saber si dos arrays tenen tots els valors idèntics. La funció rebrà dos arrays i la seva dimensió. Retornarà `True` en el cas que siguin iguals i `False` en cas contrari.
NOTA: Podeu suposar que els dos arrays tenen la mateixa dimensió.

Exercici 4 (1 punt)

Fer una funció anomenada `Escrutini` per saber si un cartró ha aconseguit Bingo. La funció rebrà un array que serà el cartró, la dimensió del cartró i un array (bombo) que contingrà informació dels noms de les boles que han sortit. En l'array `bombo` si el valor d'una determinada posició (`index`) és un 1, vindrà dir que la bola correspondent ha sortit; si el és un 0, vindrà dir que aquella bola no ha sortit.

La funció retornarà `True` en el cas que s'hagi aconseguit Bingo i `False` en cas contrari.

Exercici 6 (2,5 punts)

Suposeu que teniu els següents procediments i funcions ja implementats:

```
void Menuprincipal()
{
    cout << "---- Menu Principal ----" << endl;
    cout << "1.- Asignar cartrons" << endl;
    cout << "2.- Jugar" << endl;
    cout << "3.- Marcador" << endl;
    cout << "4.- Sortir" << endl;
}
```

Mostra per pantalla un menú amb les opcions del codi: 1. Asignar cartrons, 2. Jugar, etc.

```
void InicialitzarArray(int all, int dim, int v)
{
    int GenerarNombre(int min, int max, int a[], int dim)
    {
        // Implementació del generador
    }
}
```

Retorna un enter aleatori entre `min` i `max`, controlant que no estigui a l'array `a` de dimensió `dim`.

Fer un programa complet (declaracions globals i funció `main()`) que segueixi els següents passos:

1. Declarar (on corresponguï) les constants necessàries per tal que el programa sigui fàcilment modificable.
2. Declarar (on corresponguï) els arrays necessaris per representar els cartrons de dos jugadors anomenats `cartroplayer1` i `cartroplayer2`. Declarar també un array, anomenat `bombo`, per controlar quins nombres han sortit i quins no.
3. Utilitzar el procediment `Menuprincipal()` i llegir l'opció escollida.
4. Si l'opció és 1, generar dos cartrons (un per a cada jugador) seguint els passos següents:
 - 4.1. Inicialitzar els cartrons dels jugadors tot a 0 amb el procediment `InicialitzarArray`.
 - 4.2. Utilitzar el procediment `CrearCartro` per crear els cartrons del jugador 1 i del jugador 2.
 - 4.3. Comprovar si els dos cartrons són diferents utilitzant la funció `ArraysIguals`. En cas que siguin iguals tornar a generar un dels dos.
5. Si l'opció és 2, implementa el joc seguint els passos següents:
 - 5.1. S'haurà de comprovar que s'han generat els cartrons (s'ha entrat a l'opció 1). En cas contrari, donar un missatge d'error i tornar al menú principal sense fer res més.
 - 5.2. Inicialitzar l'array `bombo` a 0 amb el procediment `InicialitzarArray`.
 - 5.3. Generar un nombre per simular la bola amb la funció `GenerarNombre`.
 - 5.4. Imprimir el nombre que ha sortit i utilitzar la funció `ImprimirBolesSortides` per imprimir els noms de les boles que han sortit al llarg de la partida.
 - 5.5. Utilitzar la funció `Escrutini` per comprovar si algun dels dos jugadors ha aconseguit fer Bingo.
 - 5.6. En cas que no hi hagi cap guanyador tornar al punt 5.3
6. Si l'opció és 3: imprimir el marcador de les partides que s'han jugat i mostrar el missatge generat pel procediment `ImprimirComentari`.
7. Si l'opció és 4, imprimir el marcador final i mostrar el missatge: "Sortint del programa...".
8. Qualsevol altra opció, escriure el missatge: "Opció no permesa".
9. Repetir els passos 3 a 9, fins que l'opció del menú escollida sigui la 4.

Exercici 7 (0,5 punts)

Declarar un nou tipus de dades, TCartró, com un registre amb els camps: Ident_Cartró (valor enter), Numeros (array de 25 enters) i Nom_Jugador (cadena de 50 caràcters).

Fer un procediment anomenat Imprimir que rebi com a parametre un registre del tipus TCartró i imprimeix l'identificador del cartró i el nom del jugador separant-los amb el caràcter ‘,’.

Exercici 9 (1 punt)

Escriure la funció Elevar que retorna el resultat d'elevat un nombre al quadrat. La funció tindrà 2 paràmetres de tipus int. El primer serà la base i el segon serà el resultat de l'operació en el cas que es pugui realitzar. La funció retornarà un 1 si s'ha pogut realitzar l'operació i un 0 en cas contrari.

NOTA: El valor de la base ha d'estar inclos en l'interval [-46.340,46.340] per evitar overflow.

Exercici 8 (1 punt)

Fer una funció anomenada Es_MatPositiva que rebi com a paràmetres: una matriu d'enters de 4 columnes, el número de files de la matriu, i el numero de columnes de la matriu, i retorna 1 si tots els valors de la matriu són positius o zero, i un 0 en cas contrari.

Exàmens d'Informàtica (103806)

Examen Parcial (23 de Novembre de 2020)

Nom estudiant: Hector Olivé Padrós

Curs 2020-2021
Grup:
NIU: 2076813

Important: Recordeu que cal donar les millors solucions possibles en cada exercici. A més de funcionar correctament, el codi ha d'estar ben programat (codi clar, amb les instruccions més adients, sense operacions ni variables innecessàries, etc.)

Les funcions i procediments de les preguntes de la 1 a la 6 formen part d'un únic programa i, per tant, ha d'haver coherència entre les seves definicions, i la forma en què s'utilitzen en altres preguntes. El context del programa a desenvolupar, et teriu a continuació:

El Bingo de 75 boles és un joc d'aixar on hi ha 75 boles numerades de l'1 al 75 dins del bombó. Els cartrons tenen un total de 25 números. En aquest tipus de bingo, es van treient boles del bombó i es guanya un premi quan han sortit tots els números del nostre cartó. Això es coneix com a bingo o full House.

Volem fer un programa que permeti jugar al Bingo a dos jugadors. Els jugadors podran jugar tantes partides com vulguin i caldrà portar el compte de quantes partides ha guanyat cadascú.

Exercici 1 (1 punt)

Fer un procediment anomenat `CrearCartó` per a generar un cartó. S'han de generar 25 nombres del 1 al 75. S'ha d'assegurar que hi hagin:

- 5 nombres del 1'1 al 15,
- 5 nombres del 46 al 60 i
- 5 nombres del 61 al 75.

• 5 nombres del 31 al 45,
El procediment rebrà un array (que serà el cartó) i la dimensió de l'array. Retornarà l'array ple amb els números ordenats de menor a major.

UTILITZEU (no implementar) els següents procediments o funcions:

- `int GenerarNombre(int min, int max, int a[], int dim)`
Genera un nombre aleatori entre min i max, i revisa que no estigui a l'array a de dimensió dim.
- `void Ordena(int a[], int dim)`
Ordena l'array a de dimensió dim de menor a major.

$$7 \times 4 = 28$$
$$28 + 120 = \boxed{148}$$

Exercici 2 (1 punt)
Fer la funció anomenada `ArraysIguals` per saber si dos arrays tenen tots els valors idèntics.

La funció rebrà dos arrays i la seva dimensió. Retorna `true` en el cas que siguin iguals i `false` en cas contrari.

NOTA: Podeu suposar que els dos arrays tenen la mateixa dimensió.

$$800 \div 20 = 40$$

Resultat : 2

Exercici 3 (1 punt)

Fer un procediment anomenat `ImprimirBolesSortides` per imprimir els nombres de les boles que ja han sortit. El procediment rebrà un array i la seva dimensió. L'array contingrà informació sobre si un nombre ha sortit o no. Si el valor d'una determinada posició (`index`) és un 1, voldrà dir que la bola corresponent ha sortit i si el és un 0, voldrà dir que aquella bola no ha sortit.

NOTA: Recordeu que els números de les boles van de 1 a 75.

Fer una funció anomenada Escrutini per saber si un cartó ha aconseguit Bingo. La funció rebrà un array que serà el cartó, la dimensió del cartó i un array (bombo) que contindrà informació dels nombres que han sortit. En l'array bombó si el valor d'una determinada posició (index) és un 1, vindrà dir que la bola correspondent ha sortit i si el és un 0, vindrà dir que aquella bola no ha sortit.

La funció retornarà True en el cas que s'hagi aconseguit Bingo i False en cas contrari.

Exercici 4 (1 punt)

Suposeu que teniu els següents procediments i funcions ja implementats:

```
void MenuPrincipal()
{
    cout << "---- Menu Principal ----" << endl;
    cout << "1.- Asignar cartones" << endl;
    cout << "2.- Jugar" << endl;
    cout << "3.- Marcador" << endl;
    cout << "4.- Sortir" << endl;
}
```

Fer una funció anomenada ImprimirComentari per imprimir un comentari sobre el marcador. El procediment rebrà dos valors enters que seran el nombre de partides guanyades per cada jugador. En el cas que la diferència sigui:

- menor o igual que 1, escriure el missatge "Està molt igualat".
- entre 2 i 5, el missatge hauria de ser "La remuntada és possible".
- Major o igual que 6, el missatge haurà de ser "Avui és el teu dia de sort".

NOTA: podeu utilitzar la funció abs() per obtenir el valor absolut d'un nombre.

Fer un procediment anomenat ImprimirComentari per imprimir un comentari sobre el marcador. El procediment rebrà dos valors enters que seran el nombre de partides guanyades per cada jugador. En el cas que la diferència sigui:

- menor o igual que 1, escriure el missatge "Està molt igualat".
- entre 2 i 5, el missatge hauria de ser "La remuntada és possible".
- Major o igual que 6, el missatge haurà de ser "Avui és el teu dia de sort".

Fer un programa complet (declaracions globals i funció main()) que segueixi els següents passos:

1. Declarar (on corresponguï) les constants necessàries per tal que el programa sigui fàcilment modificable.
2. Declarar (on corresponguï) els arrays necessaris per representar els cartons de dos jugadors anomenats cartroPlayer1 i cartroPlayer2. Declarar també un array, anomenat bombo, per controlar quins nombres han sortit i quins no.
3. Utilitzar el procediment MenuPrincipal() i llegir l'opció escollida.
4. Si l'opció és 1, generar dos cartons (un per a cada jugador) seguint els passos següents:
 - 4.1. Inicialitzar els cartons dels jugadors tot a 0 amb el procediment InicialitzarArray.
 - 4.2. Utilitzar el procediment CrearCartro per crear els cartons del jugador 1 i del jugador 2.
 - 4.3. Comprovar si els dos cartons són diferents utilitzant la funció ArraysEqual. En cas que siguin iguals tornar a generar un dels dos.
5. Si l'opció és 2, implementa el joc seguint els passos següents:
 - 5.1. S'haurà de comprovar que s'han generat els cartons (s'ha entrat a l'opció 1). En cas contrari, donar un missatge d'error i tornar al menú principal sense fer res més.
 - 5.2. Inicialitzar l'array bombo a 0 amb el procediment InicialitzarArray.
 - 5.3. Generar un nombre per simular la bola amb la funció GenerarNombre.
 - 5.4. Imprimir el nombre que ha sortit i utilitzar la funció ImprimirBolesSortides per imprimir els nombres de les boles que han sortit al llarg de la partida.
 - 5.5. Utilitzar la funció Escrutini per comprovar si algun dels dos jugadors ha aconseguit fer Bingo.
 - 5.6. En cas que no hi hagi cap guanyador tornar al punt 5.3
 6. Si l'opció és 3, imprimir el marcador de les partides que s'han jugat i mostrar el missatge generat pel procediment ImprimirComentari.
 7. Si l'opció és 4, imprimir el marcador final i mostrar el missatge: "Sortint del programa...".
 8. Qualsevol altra opció, escriure el missatge: "Opció no permesa".
 9. Repetir els passos 3 a 9, fins que l'opció del menú escollida sigua la 4.

Exercici 7 (0.5 punts)

Declarar un nou tipus de dades, `TCartro`, com un registre amb els camps: `Ident_Cartro` (valor enter), `Numeros` (array de 25 enters) i `Nom_Jugador` (cadena de 50 caràcters).
Fer un procediment anomenat `Imprimir` que rebi com a paràmetre un registre del tipus `TCartro` i imprimeixi l'identificador del cartró i el nom del jugador separant-los amb el caràcter `:`.

Exercici 9 (1 punt)

Escriure la funció `Elevar` que retorni el resultat de elevar un nombre al quadrat. La funció tindrà dos paràmetres de tipus `int`. El primer serà la base i el segon serà el resultat de l'operació en el cas que es pugui realizar. La funció retornarà un 1 si s'ha pogut realizar l'operació i un 0 en cas contrari.

NOTA: El valor de la base ha d'estar inclos en l'intervall [-46..340,46..340] per evitar overflow.

Exercici 8 (1 punt)

Fer una funció anomenada `Es_MatPositiva` que rebi com a paràmetres: una matrícula d'enters de 4 columnes, el número de files de la matrícula i el número de columnes de la matrícula, i retorna 1 si tots els valors de la matrícula són positius o zero, i un 0 en cas contrari.

anaments d'informàtica (T03806)

Curs 2020-2021
Examen Parcial (23 de Novembre de 2020)

Nom estudiant: Nancy Stewllings (ero)

Grup: NIU: 441 83 79

Important: Recordeu que cal donar les millors solucions possibles en cada exercici. A més de funcionar correctament, el codi ha d'estar ben programat (codi clar, amb les instruccions més adients, sense operacions ni variables innecessàries, etc.)

Les funcions i procediments de les preguntes de la 1 a la 6 formen part d'un únic programa i, per tant, ha d'haver coherentia entre les seves definicions, i la forma en què s'utilitzen en altres preguntes. El context del programa a desenvolupar, el teniu a continuació:

El Bingo de 75 boles és un joc d'atzar on hi ha 75 boles numerades de l'1 al 75 dins del bumbo. Els cartells tenen un total de 25 números. En aquest tipus de bingo, es van treient boles del bumbo i es guanya un premi quan han sortit tots els nombres del nostre cartell. Això es coneix com a bingo o full House.

Volem fer un programa que permeti jugar al Bingo a dos jugadors. Els jugadors podran jugar tantes partides com vulguin i caldrà portar el compte de quantes partides ha guanyat cadascú.

Exercici 1 (1 punt)

Fer un procediment anomenat `CrearCartro` per a generar un cartell. S'han de generar 25 nombres del 1 al 75. S'ha d'assegurar que hi hagin:

- 5 nombres del 1'1 al 15,
- 5 nombres del 46 al 60 i
- 5 nombres del 61 al 75.
- 5 nombres del 16 al 30,
- 5 nombres del 31 al 45,

El procediment rebrà un array (que serà el cartell) i la dimensió de l'array. Retornarà l'array ple amb els números ordenats de menor a major.

UTILITZEU (no implementar) els següents procediments o funcions:

- `int GenerarNombre (int min, int max, int a[], int dim)`
Genera un nombre aleatori entre min i max, i revisa que no estigui a l'array a de dimensió dim.
- `void Ordena (int a[], int dim)`
Ordene l'array a de dimensió dim de menor a major.

$$5 + 14 = 23$$
$$23 \times 10 = \underline{120}$$

Fer la funció anomenada `ArraysIguals` per saber si dos arrays tenen tots els valors idèntics. La funció rebrà dos arrays i la seva dimensió. Retornarà `true` en el cas que siguin iguals i `false` en cas contrari.

NOTA: Podeu suposar que els dos arrays tenen la mateixa dimensió.

$$94 + 14 = 108$$
$$108 \div 2 = 54$$

Resultat: 54

Exercici 3 (1 punt)

Fer un procediment anomenat `ImprimirBolesSortides` per imprimir els nombres de les boles que ja han sortit. El procediment rebrà un array i la seva dimensió. L'array contingirà informació sobre si un nombre ha sortit o no. Si el valor d'una determinada posició (`index`) és un 1, voldrà dir que la bola corresponent ha sortit i si el és un 0, voldrà dir que aquella bola no ha sortit.

NOTA: Recordeu que els números de les boles van de 1 a 75.

Exercici 4 (1 punt)

Fer una funció anomenada Escrutini per saber si un cartó ha aconseguit Bingo. La funció rebrà un array que serà el cartó, la dimensió del cartó i un array (bombó) que contindrà informació dels nombres que han sortit. En l'array bombó si el valor d'una determinada posició (index) és un 1, vindrà dir que la bola correspondent ha sortit i si el és un 0, vindrà dir que aquella bola no ha sortit.

La funció retornarà True en el cas que s'hagi aconseguit Bingo i False en cas contrari.

Exercici 6 (2,5 punts)

Suposeu que teniu els següents procediments i funcions ja implementats:

```
void MenúPrincipal()
{
    cout << " --- Menú Principal --- " << endl;
    cout << " 1. - Asignar cartones " << endl;
    cout << " 2. - Jugar " << endl;
    cout << " 3. - Marcador " << endl;
    cout << " 4. - Sortir " << endl;
}
```

Mostra per pantalla un menú amb les opcions del codi: 1. Assignar cartons, 2. Jugar, etc.

void InicialitzarArray(int a[], int dim, int v)

Inicialitza el vector a de dimensió dim al valor v

int GenerarNombre(int min, int max, int a[], int dim)

Retorna un enter aleatori entre min i max, controlant que no estigui a l'array a de dimensió dim

Fer un programa complet (declaracions globals i funció main()) que segueixi els següents passos:

1. Declarar (on correspongui) les constants necessàries per tal que el programa sigui fàcilment modificable.
2. Declarar (on correspongui) els arrays necessaris per representar els cartons de dos jugadors anomenats cartroPlayer1 i cartroPlayer2. Declarar també un array, anomenat bombó, per controlar quins nombres han sortit i quins no.
3. Utilitzar el procediment MenúPrincipal() i llegir l'opció escollida.
4. Si l'opció és 1, generar dos cartons (un per a cada jugador) seguint els passos següents:
 - 4.1. Inicialitzar els cartons dels jugadors tot a 0 amb el procediment InicialitzarArray.
 - 4.2. Utilitzar el procediment CrearCartó per crear els cartons del jugador 1 i del jugador 2.
- 4.3. Comprovar si els dos cartons són diferents utilitzant la funció ArraysIguals. En cas que siguin iguals tornar a generar un dels dos.
5. Si l'opció és 2, implementa el joc seguint els passos següents:
 - 5.1. S'haurà de comprovar que s'han generat els cartons (s'ha entrat a l'opció 1). En cas contrari, donar un missatge d'error i tornar al menú principal sense fer res més.
 - 5.2. Inicialitzar l'array bombó a 0 amb el procediment InicialitzarArray.
 - 5.3. Generar un nombre per simular la bola amb la funció GenerarNombre.
 - 5.4. Imprimir el nombre que ha sortit i utilitzar la funció ImprimirBolesSortides per imprimir els nombres de les boles que han sortit al llarg de la partida.
 - 5.5. Utilitzar la funció Escrutini per comprovar si algun dels dos jugadors ha aconseguit fer Bingo.
 - 5.6. En cas que no hi hagi cap guanyador tornar al punt 5.3
 6. Si l'opció és 3, imprimir el marcador de les partides que s'han jugat i mostrar el missatge generat pel procediment ImprimirComentari.
 7. Si l'opció és 4, imprimir el marcador final i mostrar el missatge: "Sortint del programa . . .".
 8. Qualsevol altra opció, escriure el missatge: "Opció no permesa".
 9. Repetir els passos 3 a 9, fins que l'opció del menú escollida sigui la 4.

Exercici 7 (0,5 punts)
Declarar un nou tipus de dades, `TCarro`, com un registre amb els camps: `Ident_Carro` (valor enter), `Numeros` (array de 25 enters) i `nom_Jugador` (cadena de 50 caràcters).

Fer un procediment anomenat `Imprimir` que rebi com a paràmetre un registre del tipus `TCarro` i imprimeixi l'identificador del carro i el nom del jugador separant-los amb el caràcter '-'.

Exercici 9 (1 punt)

Escriure la funció `Elevar` que retorni el resultat de elevar un nombre al quadrat. La funció tindrà «paràmetres de tipus int». El primer serà la base i el segon serà el resultat de l'operació en el cas que es pugui realizar. La funció retornarà un 1 si s'ha pogut realizar l'operació i un 0 en cas contrari.

NOTA: El valor de la base ha de estar inclos en l'intervall [-46.340, 46.340] per evitar overflow.

Exercici 8 (1 punt)

Fer una funció anomenada `E.S_MatPositiva` que rebi com a paràmetres: una matriu d'enters de 4 columnes, el número de files de la matriu, i el número de columnes de la matriu, i retorna 1 si tots els valors de la matriu són positius o zero, i un 0 en cas contrari.

Grup: 5269877
NIU:

Fer la funció anomenada `ArraysIguals` per saber si dos arrays tenen tots els valors idèntics. La funció rebrà dos arrays i la seva dimensió. Retornarà `True` en el cas que siguin iguals i `False` en cas contrari.

Important: Recordeu que cal donar les millors solucions possibles en cada exercici. A més de funcionar correctament, el codi ha d'estar ben programat (codi clar, amb les instruccions més adients, sense operacions ni variables innecessàries, etc.)

Les funcions i procediments de les preguntes de la 1 a la 6 formen part d'un únic programa i, per tant, ha d'haver coherència entre les seves definicions, i la forma en què s'utilitzen en altres preguntes. El context del programa a desenvolupar, el tenui a continuació:

El Bingo de 75 boles és un joc d'atzar on hi ha 75 boles numerades de l'1 al 75 dins del bombo. Els cartons tenen un total de 25 nombres. En aquest tipus de bingo, es van treient boles del bombo i es guanya un premi quan han sortit tots els nombres del nostre cartó. Això es coneix com a bingo o full House.

Volem fer un programa que permeti jugar al Bingo a dos jugadors. Els jugadors podran jugar tantes partides com vulguin i caindrà portar el compte de quantes partides ha guanyat cadascú.

Exercici 1 (1 punt)

Fer un procediment anomenat `CrearCartiro` per a generar un cartó. S'han de generar 25 números del 1 al 75. S'ha d'assegurar que hi hagin:

- 5 nombres del 11 al 15,
- 5 nombres del 46 al 60 i

- 5 nombres del 61 al 75.

- 5 nombres del 31 al 45,

El procediment rebrà un array (que serà el cartó) i la dimensió de l'array. Retornarà l'array ple amb els números ordenats de menor a major.

UTILITZEU (no implementar) els següents procediments o funcions:

- `int GenerarNombre (int min, int max, int a[], int dim)`
Genera un nombre aleatori entre min,max, i revisa que no estigui a l'array a de dimensió dim.
- `void Ordena (int a[], int dim)`
Ordena l'array a de dimensió dim de menor a major.

$$14 + 4 = \underline{\underline{18}}$$

$$18 \times 4 = \underline{\underline{72}}$$

Exercici 4 (1 punt)

Fer una funció anomenada `Escrutini` per saber si un cartó ha aconseguit Bingo. La funció rebrà un array que serà el cartó, la dimensió del cartó i un array (bombo) que contingrà informació dels nombres que han sortit. En l'array bombo si el valor d'una determinada posició (`index`) és un 1, vindrà dir que la bola correspondent ha sortit i si el és un 0, vindrà dir que aquella bola no ha sortit.

La funció retornarà `True` en el cas que s'hagi aconseguit Bingo i `False` en cas contrari.

Exercici 5 (2.5 punts)

Suposeu que teniu els següents procediments i funcions ja implementats:

```
void Menuprincipal()
{
    cout << " --- Menu Principal --- " << endl;
    cout << "1.- Jugar" << endl;
    cout << "3.- Marcador" << endl;
    cout << "4.- Sortir" << endl;
}
```

Mostra per pantalla un menú amb les opcions del codi: 1. Assignar cartons, 2. Jugar, etc.

void InicialitzazzArray(int a[], int dim, int v)

Inicialitza el vector a de dimensió dim al valor v

int GenerarNombre (int min, int max, int a[], int dim)

Retorna un enter aleatori entre min i max, controlant que no estigui a l'array a de dimensió dim

Fer un programa complet (declaracions globals i funció main()) que segueixi els següents passos:

1. Declarar (on corresponguï) les constants necessàries per tal que el programa sigui fàcilment modificable.
2. Declarar (on corresponguï) els arrays necessaris per representar els cartons de dos jugadors anomenats `cartroPlayer1` i `cartroPlayer2`. Declarar també un array anomenat `bombo`, per controlar quins nombres han sortit i quins no.
3. Utilitzar el procediment `Menuprincipal()` i llegir l'opció escollida.
4. Si l'opció és 1, generar dos cartons (un per a cada jugador) seguint els passos següents:
 - 4.1. Inicialitzar els cartons dels jugadors tot a 0 amb el procediment `InicialitzarArray`.
 - 4.2. Utilitzar el procediment `CrearCartro` per crear els cartons del jugador 1 i del jugador 2.
- 4.3. Comprovar si els dos cartons són diferents utilitzant la funció `ArraysIguals`. En cas que siguin iguals tornar a generar un dels dos.
5. Si l'opció és 2, implementa el joc seguint els passos següents:
 - 5.1. S'haurà de comprovar que s'han generat els cartons (s'ha entrat a l'opció 1). En cas contrari, donar un missatge d'error i tornar al menú principal sense fer res més.
 - 5.2. Inicialitzar l'array `bombo` a 0 amb el procediment `InicialitzarArray`.
 - 5.3. Generar un nombre per simular la bola amb la funció `GenerarNombre`.
 - 5.4. Imprimir el nombre que ha sortit i utilitzar la funció `ImprimirBolesSortides` per imprimir els nombres de les botes que han sortit al llarg de la partida.
 - 5.5. Utilitzar la funció `Escrutini` per comprovar si algun dels dos jugadors ha aconseguit fer Bingo.
- 5.6. En cas que no hi hagi cap guanyador tornar al punt 5.3
6. Si l'opció és 3, imprimir el marcador de les partides que s'han jugat i mostrar el missatge generat pel procediment `ImprimirComentari`.
7. Si l'opció és 4, imprimir el marcador final i mostrar el missatge: "Sortint del programa . . ."
8. Qualsevol altra opció, escriure el missatge: "Opció no permesa".
9. Repetir els passos 3 a 9, fins que l'opció del menú escollida sigui la 4.

Exercici 7 (0.5 punts)

Declarar un nou tipus de dades, `TCartro`, com un registre amb els camps: `Ident_Cartro` (valor enter), `Numeros` (array de 25 enters) i `nom_Jugador` (cadena de 50 caràcters).

Fer un procediment anomenat `Imprimir` que rebi com a paràmetre un registre del tipus `TCartro` i imprimeixi l'identificador del cartró i el nom del jugador separant-los amb el caràcter '`,`'.

Exercici 9 (1 punt)

Escriviu la funció `Elevar` que retorni el resultat d'elevar un nombre al quadrat. La funció tindrà dos paràmetres de tipus `int`. El primer serà la base i el segon serà el resultat de l'operació en el cas que es pugui realitzar. La funció retornarà un 1 si s'ha pogut realizar l'operació i un 0 en cas contrari.

NOTA: El valor de la base ha d'estar inclos en l'interval [-46.340, 46.340] per evitar overflow.

Exercici 8 (1 punt)

Fer una funció anomenada `Es_MatPositiva` que rebi com a paràmetres: una matriu d'enters de 4 columnes, el número de files de la matriu, i el número de columnes de la matriu, i retorna 1 si tots els valors de la matriu són positius o zero, i un 0 en cas contrari.

Fer la funció anomenada `ArraysIguals` per saber si dos arrays tenen tots els valors idèntics. La funció rebrà dos arrays i la seva dimensió. Retornarà `True` en el cas que siguin iguals i `False` en cas contrari.

NOTA: Podeu suposar que els dos arrays tenen la mateixa dimensió.

Important: Recordeu que cal donar les millors solucions possibles en cada exercici. A més de funcionar correctament, el codi ha d'estar ben programat (codi clar, amb les instruccions més adients, sense operacions ni variables innecessàries, etc.)

Les funcions i procediments de les preguntes de la 1 a la 6 formen part d'un únic programa i, per tant, ha d'haver coherentia entre les seves definicions, i la forma en què s'utilitzen en altres preguntes. El context del programa a desenvolupar, el teniu a continuació:

El Bingo de 75 boles és un joc d'atzar on hi ha 75 boles numerades de l'1 al 75 dins del bombo. Els cartolls tenen un total de 25 nombres. En aquest tipus de bingo, es van treient boles del bombo i es guanya un premi quan han sortit tots els nombres del nostre cartell. Això es coneix com a bingo o full House.

Volem fer un programa que permeti jugar al Bingo a dos jugadors. Els jugadors podran jugar tantes partides com vulguin i caindrà portar el compte de quantes partides ha guanyat cadaescú.

Exercici 1 (1 punt)

Fer un procediment anomenat `CrearCartell` per a generar un cartell. S'han de generar 25 nombres del 1 al 75. S'ha d'assegurar que hi hagin:

- 5 nombres del 1 al 15,
- 5 nombres del 16 al 30,
- 5 nombres del 31 al 45,
- 5 nombres del 46 al 60 i
- 5 nombres del 61 al 75.

El procediment rebrà un array (que serà el cartell) i la dimensió de l'array. Retornarà l'array ple amb els números ordenats de menor a major.

UTILITZEU (no implementar) els següents procediments o funcions:

- `int GenerarNombre(int min, int max, int aff, int dim)`
Genera un nombre aleatori entre min i max. i revisa que no estigui a l'array a de dimensió dim.
- `void Ordena (int a[], int dim)`
Ordena l'array a de dimensió dim de menor a major.

$$31 + 48 = \underline{79}$$
$$79 + 86 = \underline{165}$$

Resultat: 76

$$56 \times 1 = 56$$
$$20 + 56 = 76$$

Exercici 3 (1 punt)

Fer un procediment anomenat `ImprimirBolesSortides` per imprimir els nombres de les boles que ja han sortit. El procediment rebrà un array i la seva dimensió. L'array contingirà informació sobre si un nombre ha sortit o no. Si el valor d'una determinada posició (`index`) és un 1, voldrà dir que la bola corresponent ha sortit i si el és un 0, voldrà dir que aquella bola no ha sortit.

NOTA: Recordeu que els números de les boles van de 1 a 75.

Exercici 4 (1 punt)

Fer una funció anomenada `Escrutini` per saber si un cartró ha aconseguit Bingo. La funció rebrà un array que serà el cartró, la dimensió del cartró i un array (bombo) que contindrà informació dels nombres que han sortit. En l'array `bombo` si el valor d'una determinada posició (`index`) és un 1, vindrà dir que la bola correspondent ha sortit i si el és un 0, vindrà dir que aquella bola no ha sortit.

La funció retornarà `true` en el cas que s'hagi aconseguit Bingo i `false` en cas contrari.

Exercici 6 (2.5 punts)

Suposeu que teniu els següents procediments i funcions ja implementats:

```
void MenuPrincipal()
{
    cout << "---- Menu Principal ----" << endl;
    cout << "1.- Asignar cartrons" << endl;
    cout << "2.- Jugar" << endl;
    cout << "3.- Marcador" << endl;
    cout << "4.- Sortir" << endl;
}
```

Mostra per pantalla un menú amb les opcions del codi: 1. Asignar cartrons, 2. Jugar, etc.

```
void InicialitzarArray(int a[], int dim, int v)
{
    int GenerarNombre(int min, int max, int a[], int dim)
    {
        Retorna un enter aleatori entre min i max, controlant que no estigui a l'array a de dimensió dim
    }
}
```

Inicialitza el vector `a` de dimensió `dim` al valor `v`.

Fer un programa complet (declaracions globals i funció `main()`) que segueixi els següents passos:

1. Declarar (on correspongui) les constants necessàries per tal que el programa sigui fàcilment modificable.
2. Declarar (on correspongui) els arrays necessaris per representar els cartrons de dos jugadors anomenats `cartroPlayer1` i `cartroPlayer2`. Declarar també un array, anomenat `bombo`, per controlar quins nombres han sortit i quins no.
3. Utilitzar el procediment `MenuPrincipal()` i llegir l'opció escollida.
4. Si l'opció és 1, general dos cartrons (un per a cada jugador) seguint els passos següents:
 - 4.1. Inicialitzar els cartrons dels jugadors tot a 0 amb el procediment `InicialitzarArray`.
 - 4.2. Utilitzar el procediment `CrearCartro` per crear els cartrons del jugador 1 i del jugador 2.
- 4.3. Comprovar si els dos cartrons són diferents utilitzant la funció `ArraysIguals`. En cas que siguin iguals tornarà a generar un dels dos.
5. Si l'opció és 2, implementa el joc seguint els passos següents:
 - 5.1. S'haurà de comprovar que s'han generat els cartrons (s'ha entrat a l'opció 1). En cas contrari, donar un missatge d'error i tornar al menú principal sense fer res més.
 - 5.2. Inicialitzar l'array `bombo` a 0 amb el procediment `InicialitzarArray`.
 - 5.3. Generar un nombre per simular la bola amb la funció `GenerarNombre`.
 - 5.4. Imprimir el nombre que ha sortit i utilitzar la funció `ImprimirBolesSortides` per imprimir els nombres de les boles que han sortit al llarg de la partida.
 - 5.5. Utilitzar la funció `Escrutini` per comprovar si algun dels dos jugadors ha aconseguit fer Bingo.
 - 5.6. En cas que no hi hagi cap guanyador tornar al punt 5.3
6. Si l'opció és 3: imprimir el marcador de les partides que s'han jugat i mostrar el missatge generat pel procediment `ImprimirComentari`.
7. Si l'opció és 4, imprimir el marcador final i mostrar el missatge: "Sortint del programa..."
8. Qualsevol altra opció, escriure el missatge: "Opció no permesa".
9. Repetir els passos 3 a 9, fins que l'opció del menú escollida sigui la 4.

Exercici 7 (0.5 punts)

Declarar un nou tipus de dades, `TCartro`, com un registre amb els camps: `Ident_Cartro` (valor enter), `Numeros` (array de 25 enters) i `Nom_Jugador` (cadena de 50 caràcters).
Fer un procediment anomenat `Imprimir` que rebi com a paràmetre un registre del tipus `TCartro` i imprimeixi l'identificador del cartró i el nom del jugador separant-los amb el caràcter `'.'`.

Exercici 9 (1 punt)

Escrivre la funció `Elevar` que retorni el resultat de elevar un nombre al quadrat. La funció tindrà parametres de tipus `int`. El primer serà la base i el segon serà el resultat de l'operació en el cas que es pugui realizar. La funció retornarà un 1 si s'ha pogut realizar l'operació i un 0 en cas contrari.

NOTA: El valor de la base ha de estar inclos en l'interval [-46.340,46.340] per evitar overflow.

Exercici 8 (1 punt)

Fer una funció anomenada `E8_MatPositiva` que rebi com a paràmetres: una matriu d'enters de 4 columnes, el número de files de la matriu, i el número de columnes de la matriu, i retorna 1 si tots els valors de la matriu són positius o zero, i un 0 en cas contrari.

Important: Recordeu que cal donar les millors solucions possibles en cada exercici. A més de funcionar correctament, el codi ha d'estar ben programat (codi clar, amb les instruccions més adients, sense operacions ni variables innecessàries, etc.)

Les funcions i procediments de les preguntes de la 1 a la 6 formen part d'un únic programa i, per tant, ha d'haver coherentia entre les seves definicions, i la forma en què s'utilitzen en altres preguntes. El context del programa a desenvolupar, el teniu a continuació:

El Bingo de 75 boles és un joc d'atzar on hi ha 75 boles numerades de l'1 al 75 dins del bumbo. Els cartells tenen un total de 25 nombres. En aquest tipus de bingo, es van treient boles del bumbo i es guanya un premi quan han sortit tots els nombres del nostre cartell. Això es coneix com a bingo o full House.

Volem fer un programa que permeti jugar al Bingo a dos jugadors. Els jugadors podran jugar tantes partides com vulguin i caldrà portar el compte de quantes partides ha guanyat cadascú.

Exercici 1 (1 punt)

Fer un procediment anomenat `CrearCartell` per a generar un cartell. S'han de generar 25 nombres del 1 al 75. S'ha d'assegurar que hi hagin:

- 5 nombres del 1 al 15,
- 5 nombres del 16 al 30,
- 5 nombres del 31 al 45,
- 5 nombres del 46 al 60 i
- 5 nombres del 61 al 75.

El procediment rebrà un array (que serà el cartell) i la dimensió de l'array. Retornarà l'array ple amb els números ordenats de menor a major.

UTILIZEU (no implementar) els següents procediments o funcions:

- `int GenerarNombre (int min, int max, int a[], int dim)`
Genera un nombre aleatori entre min i max, i revisa que no estigui a l'array a de dimensió dim.
- `void Ordena (int a[], int dim)`
Ordena l'array a de dimensió dim de menor a major.

$$10 \times 10 = \underline{100}$$

$$100 + 300 = \underline{400}$$

Fer la funció anomenada `ArraysIguals` per saber si dos arrays tenen tots els valors idèntics. La funció rebrà dos arrays i la seva dimensió. Retornarà `true` en el cas que siguin iguals i `false` en cas contrari.

NOTA: Podeu suposar que els dos arrays tenen la mateixa dimensió.

$$999 - 333 = 666$$

$$666 + 333 = 999$$

Resultat : 999

Exercici 3 (1 punt)

Fer un procediment anomenat `ImprimirBolesSortides` per imprimir els nombres de les boles que ja han sortit. El procediment rebrà un array i la seva dimensió. L'array contingrà informació sobre si un nombre ha sortit o no. Si el valor d'una determinada posició (`index`) és un 1, voldrà dir que la bola correspondent ha sortit i si el és un 0, voldrà dir que aquella bola no ha sortit.

NOTA: Recordeu que els números de les boles van de 1 a 75.

Fer una funció anomenada Escrutini per saber si un cartó ha aconseguit Bingo. La funció rebrà un array que serà el cartó, la dimensió del cartó i un array (bombo) que contindrà informació dels nombres que han sortit. En l'array bombo si el valor d'una determinada posició (índex) és un 1, vindrà dir que la bola correspondent ha sortit i si el és un 0, vindrà dir que aquella bola no ha sortit.

La funció retornarà true en el cas que s'hagi aconseguit Bingo i false en cas contrari.

Exercici 6 (2,5 punts)
Suposeu que teniu els següents procediments i funcions ja implementats:

```
void Menuprincipal()
{
    cout << " --- Menu Principal --- " << endl;
    cout << "1. - Asignar cartrons " << endl;
    cout << "2. - Jugar" << endl;
    cout << "3. - Marcador" << endl;
    cout << "4. - Sortir" << endl;
}
```

Mostra per pantalla un menú amb les options del codi: 1. Assignar cartrons, 2. Jugar, etc.

```
void InicialitzarArray(int a[], int dim, int v)
Inicialitza el vector a de dimensió dim al valor v
```

Retorna un enter aleatori entre min i max, controlant que no estigui a l'array a de dimensió dim

Fer un programa complet (declaracions globals i funció main()) que segueixi els següents passos:

1. Declarar (on correspongui) les constants necessàries per tal que el programa sigui fàcilment modificable.
2. Declarar (on correspongui) els arrays necessaris per representar els cartons de dos jugadors anomenats cartroPlayer1 i cartroPlayer2. Declarar també un array, anomenat bombó, per controlar quins nombres han sortit i quins no.
3. Utilitzar el procediment Menuprincipal() i llegir l'opció escollida.
4. Si l'opció és 1, generar dos cartons (un per a cada jugador) seguint els passos següents:
 - 4.1. Inicialitzar els cartons dels jugadors tot a 0 amb el procediment InicialitzarArray.
 - 4.2. Utilitzar el procediment CrearCartro per crear els cartons del jugador 1 i del jugador 2.
 - 4.3. Comprovar si els dos cartons són diferents utilitzant la funció ArraysIguals. En cas que siguin iguals tornar a generar un dels dos.
5. Si l'opció és 2, implementa el joc següint els passos següents:
 - 5.1. Shaurà de comprovar que s'han generat els cartons (s'ha entrat a l'opció 1). En cas contrari, donar un missatge d'error i tornar al menú principal sense fer res més.
 - 5.2. Inicialitzar l'array bombo a 0 amb el procediment InicialitzarArray.
 - 5.3. Generar un nombre per simular la bola amb la funció GenerarNombre.
 - 5.4. Imprimir el nombre que ha sortit i utilitzar la funció ImprimirBolesSortides per imprimir els nombres de les botes que han sortit al llarg de la partida.
 - 5.5. Utilitzar la funció Escrutini per comprovar si algun dels dos jugadors ha aconseguit fer Bingo.
 - 5.6. En cas que no hi hagi cap guanyador tornar al punt 5.3
 6. Si l'opció és 3: imprimir el marcador de les partides que s'han jugat i mostrar el missatge generat pel procediment ImprimirComentari.
 7. Si l'opció és 4, imprimir el marcador final i mostrar el missatge: "Sortint del programa....".
 8. Qualsevol altra opció, escriure el missatge: "Opció no permesa".
 9. Repetir els passos 3 a 9, fins que l'opció del menú escollida sigui la 4.

Declarar un nou tipus de dades, TCartró, com un registre amb els camps: Ident_Cartró (valor enter), Numeros (array de 25 enters) i Nom_Jugador (cadena de 50 caràcters). Fer un procediment anomenat Imprimir que rebi com a paràmetre un registre del tipus TCartró i imprimeix l'identificador del cartró i el nom del jugador separant-los amb el caràcter ‘_’.

Exercici 9 (1 punt)
Escriure la funció Elevar que retorni el resultat d'elevat un nombre al quadrat. La funció tindrà dos paràmetres de tipus int. El primer serà la base i el segon serà el resultat de l'operació en el cas que es pugui realizar. La funció retornarà un 1 si s'ha pogut realizar l'operació i un 0 en cas contrari.

NOTA: El valor de la base ha d'estar inclòs en l'intervall [-46.340,46.340] per evitar overflow.

Exercici 8 (1 punt)

Fer una funció anomenada Es_MatPositiva que rebi com a paràmetres: una matriu d'enters de 4 columnes, el número de files de la matriu, i el número de columnes de la matriu. i retorna 1 si tots els valors de la matriu són positius o zero, i un 0 en cas contrari.

Important: Recordeu que cal donar les millors solucions possibles en cada exercici. A més de funcionar correctament, el codi ha d'estar ben programat (codi clar, amb les instruccions més adients, sense operacions ni variables innecessàries, etc.)

Les funcions i procediments de les preguntes de la 1 a la 6 formen part d'un únic programa i, per tant, ha d'haver coherentia entre les seves definicions, i la forma en què s'utilitzen en altres preguntes. El context del programa a desenvolupar, el teniu a continuació:

El Bingo de 75 boles és un joc d'atzar on hi ha 75 boles numerades de l'1 al 75 dins del bombo. Els cartells tenen un total de 25 nombres. En aquest tipus de bingo, es van treient boles del bombo i es guanya un premi quan han sortit tots els nombres del nostre cartell. Això es coneix com a bingo o full House.

Volem fer un programa que permeti jugar al Bingo a dos jugadors. Els jugadors podran jugar tantes partides com vulguin i caldrà portar el compte de quantes partides ha guanyat cadascú.

Exercici 1 (1 punt)
Fer un procediment anomenat `CrearCartell` per a generar un cartell. S'han de generar 25 nombres del 1 al 75. S'ha d'assegurar que hi hagin:

- 5 nombres del 1 al 15,
- 5 nombres del 16 al 30,
- 5 nombres del 46 al 60 i
- 5 nombres del 61 al 75.

El procediment rebrà un array (que serà el cartell) i la dimensió de l'array. Retornarà l'array ple amb els números ordenats de menor a major.

UTILITZEU (no implementar) els següents procediments o funcions:

- `int GenerarNombre (int min, int max, int a[], int dim)`
Genera un nombre aleatori entre min,max, i revisa que no estigui a l'array a de dimensió dim.
- `void Ordena (int a[], int dim)`
Ordene l'array a de dimensió dim de menor a major.

$$239 - 100 = 139$$

$$239 + 100 = \boxed{339}$$

La funció rebrà dos arrays i la seva dimensió. Retornarà true en el cas que siguin iguals i false en cas contrari.

NOTA: Podeu suposar que els dos arrays tenen la mateixa dimensió.

$$326 \times 2 = 625$$

$$\text{Resultat: } 777$$

Exercici 3 (1 punt)

Fer un procediment anomenat `ImprimirBolesSortides` per imprimir els nombres de les boles que ja han sortit. El procediment rebrà un array i la seva dimensió. L'array contingrà informació sobre si un nombre ha sortit o no. Si el valor d'una determinada posició (index) és un 1, voldrà dir que la bola corresponent ha sortit i si el és un 0, voldrà dir que aquella bola no ha sortit.

NOTA: Recordeu que els números de les boles van de 1 a 75.

Fer una funció anomenada `Escrutini`, per saber si un cartró ha aconseguit Bingo. La funció rebrà un array que serà el cartró, la dimensió del cartró i un array (bombo) que contingrà informació dels noms que han sortit. En l'array bombó si el valor d'una determinada posició (`index`) és 1, vindrà dir que la bola correspondent ha sortit i si el és un 0, vindrà dir que aquella bola no ha sortit.

La funció retornarà `True` en el cas que s'hagi aconseguit Bingo i `false` en cas contrari.

```
Suposeu que teniu els següents procediments i funcions ja implementats:
void Menuprincipal()
{
    cout << " --- Menu Principal --- " << endl;
    cout << "1.- Asignar cartrons " << endl;
    cout << "2.- Jugar" << endl;
    cout << "3.- Marcador" << endl;
    cout << "4.- Sortir" << endl;
}
```

Mostra per pantalla un menú amb les opcions del codi: 1. Assignar cartrons, 2. Jugar, etc.

```
void InicialitzarArray(int a[], int dim, int v)
```

Inicialitza el vector a de dimensió dim al valor v

```
int GenerarNombre(int min, int max, int a[], int dim)
```

Retorna un enter aleatori entre min i max, controlant que no estigui a l'array a de dimensió dim

Fer un programa complet (declaracions globals i funció main()) que segueixi els següents passos:

- Declarar (o corresponguï) les constants necessàries per tal que el programa sigui fàcilment modificable.
- Declarar (o corresponguï) els arrays necessaris per representar els cartrons de dos jugadors anomenats `cartroplayer1` `cartroplayer2`. Declarar també un array, anomenat `bombo`, per controlar quins nombres han sortit i quins no.
- Utilitzar el procediment `Menuprincipal()` i llegir l'opció escollida.
- Si l'opció és 1, generar dos cartrons (un per a cada jugador) seguint els passos següents:
 - Inicialitzar els cartrons dels jugadors tot a 0 amb el procediment `InicialitzarArray`.
 - Utilitzar el procediment `CrearCartro` per crear els cartrons del jugador 1 i del jugador 2.
 - Comprovar si els dos cartrons són diferents utilitzant la funció `ArraysIguals`. En cas que siguin iguals tornar a generar un dels dos.
- Si l'opció és 2, implementa el joc seguint els passos següents:
 - Shaurà de comprovar que s'han generat els cartrons (s'ha entrat a l'opció 1). En cas contrari, donar un missatge d'error i tornar al menu principal sense fer res més.
 - Inicialitzar l'array `bombo` a 0 amb el procediment `InicialitzarArray`.
 - Generar un nombre per simular la bola amb la funció `GenerarNombre`.
 - Imprimir el nombre que ha sortit i utilitzar la funció `ImprimirBolesSortides` per imprimir els noms de les boles que han sortit al llarg de la partida.
 - Utilitzar la funció `Escrutini` per comprovar si algun dels dos jugadors ha aconseguit fer Bingo.
 - En cas que no hi hagi cap guanyador tornar al punt 5.3
 - Si l'opció és 3: imprimir el marcador de les partides que s'han jugat i mostrar el missatge generat pel procediment `ImprimirComentari`.
 - Si l'opció és 4, imprimir el marcador final i mostrar el missatge: "Sortint del programa..."
 - Qualsevol altra opció, escriure el missatge: "Opció no permesa".
 - Repetir els passos 3 a 9, fins que l'opció del menú escollida sigui la 4.

Declarar un nou tipus de dades, `TCartro`, com un registre amb els camps: `Ident_Cartro` (valor enter), `Numeros` (array de 25 enters) i `Nom_Jugador` (cadena de 50 caràcters).

Fer un procediment anomenat `Imprimir` que rebi com a paràmetre un registre del tipus `TCartro` i imprimeix l'identificador del cartó i el nom del jugador separant-los amb el caràcter ‘.

Escrive la funció `Elevat` que retorna el resultat d'elevant un nombre al quadrat. La funció té paràmetres de tipus int. El primer serà la base i el segon serà el resultat de l'operació en el cas que es pugui realizar. La funció retornarà un 1 si s'ha pogut realitzar l'operació i un 0 en cas contrari.

NOTA: El valor de la base ha d'estar inclos en l'intervall [-46,340,46,340] per evitar overflow.

Exercici 8 (1 punt)

Fer una funció anomenada `E_S_MatPositiva` que rebi com a paràmetres: una matriu d'enters de 4 columnes, el número de files de la matriu, i el número de columnes de la matriu, i retorna 1 si tots els valors de la matriu són positius o zero, i un 0 en cas contrari.

Examen Parcial (23 de Noviembre de 2020)

Grupo:
Nombre estudiante: _____ NIU: 2158560

Importante: Recuerda que hay que dar las mejores soluciones posibles en cada ejercicio. Además de funcionar correctamente, el código debe estar bien programado (código claro, con las instrucciones más adecuadas, sin operaciones ni variables innecesarias, etc.)

Las funciones y procedimientos de las preguntas de la 1 a la 6 forman parte de un único programa y, por tanto, debe haber coherencia entre sus definiciones, y la forma en que se utilizan en otras preguntas. El contexto del programa a desarrollar, lo tenéis a continuación:

El Bingo de 75 bolas es un juego de azar donde hay 75 bolas numeradas del 1 al 75 dentro del bombo. Los cartones tienen un total de 25 números. En este tipo de bingo, se van sacando bolas del bombo y se gana un premio cuando han salido todos los números de nuestro cartón. Esto se conoce como bingo o full House.

Queremos hacer un programa que permita jugar al Bingo a dos jugadores. Los jugadores podrán jugar tantas partidas como quieran y habrá que llevar la cuenta de cuántas partidas ha ganado cada uno.

Ejercicio 1 (1 punto)

Hacer un procedimiento llamado `CrearCarton` para generar un cartón. Se deben generar 25 números del 1 al 75. Se debe asegurar que haya:

- 5 números del 1 al 15,
- 5 números del 16 al 30,
- 5 números del 31 al 45,
- 5 números del 46 al 60 y
- 5 números del 61 al 75.

el procedimiento recibirá un array (que será el cartón) y la dimensión del array. Devolverá el array lleno con los números ordenados de menor a mayor.

UTILIZA (no implementar) los siguientes procedimientos o funciones:

- `int GenerarNombre (int min, int max, int a [], int dim)`
Genera un número aleatorio entre min y max, y revisa que no esté en el array a de dimensión dim.
- `void Ordenar (int a [], int dim)`
Ordenar el array a de dimensión dim de menor a mayor.

$$50 \times 7 = 350$$

$$350 - 63 = \boxed{287}$$

Ejercicio 2 (1 punto)
Hacer la función llamada `ArraysIguales` para saber si dos arrays tienen todos los valores idénticos. La función recibirá dos arrays y su dimensión, devolverá `true` en caso de que sean iguales y `false` en caso contrario.

NOTA: Podéis suponer que los dos arrays tienen la misma dimensión.

$$5 \times 8 = 40$$

$$40 + 68 = 108$$

Resultado: 108

Ejercicio 3 (1 punto)

Hacer un procedimiento llamado `ImprimirBolesSortides` para imprimir los números de las bolas que ya han salido. El procedimiento recibirá un array y su dimensión. El array contendrá información sobre si un número ha salido o no. Si el valor de una determinada posición (índice) es un 1, querrá decir que la bola correspondiente ha salido y si es un 0, querrá decir que aquella bola no ha salido.

NOTA: Recuerda que los números de las bolas van de 1 a 75.

Ejercicio 4 (1 punto)

Hacer una función llamada Escrutinio para saber si un cartón ha conseguido Bingo. La función recibirá un array que será el cartón, la dimensión del cartón y un array (bombo) que contendrá información de los números que han salido. En el array bombó si el valor de una determinada posición (índice) es un 1, querrá decir que la bola correspondiente ha salido y si es un 0, querrá decir que aquella bola no ha salido.

La función devolverá true en caso de que se haya conseguido Bingo y false en caso contrario.

Ejercicio 6 (2.5 puntos)

Suponga que tiene los siguientes procedimientos y funciones ya implementados:

```
void MenúPrincipal ()  
{ cout << "1.- Asignar cartones" << endl;  
cout << "2.- Jugar" << endl;  
cout << "3.- Marcador" << endl;  
cout << "4.- Salir" << endl; }
```

Muestra por pantalla un menú con las opciones del código: 1. Asignar cartones, 2. Jugar, etc.

```
void InicializarArray (int min, int max, int a [], int dim)  
Iniciaiza el vector a de dimensión dim al valor v
```

```
int GenerarNombre (int min, int max, int a [], int dim)
```

Devuelve un entero aleatorio entre min y max, controlando que no esté en el array a de dimensión dim

Hacer un programa completo (declaraciones globales y función main ()) que siga los siguientes pasos:

1. Declarar (donde corresponda) las constantes necesarias para que el programa sea fácilmente modificable.
2. Declarar (donde corresponda) los arrays necesarios para representar los cartones de dos jugadores llamados cartonPlayer1 y cartonPlayer2. Declarar también un array, llamado bombo, para controlar qué números han salido y cuáles no.
3. Utilizar el procedimiento MenuPrincipal () y leer la opción elegida.
4. Si la opción es 1, generar dos cartones (uno para cada jugador) siguiendo los pasos siguientes:
 - 4.1. Inicializar los cartones de los jugadores todo a 0 con el procedimiento InicializarArray.
 - 4.2. Utilizar el procedimiento CrearCartón para crear los cartones del jugador 1 y del jugador 2.
5. Si la opción es 2, comprobar si los dos cartones son diferentes utilizando la función ArraysIguals. En caso de que sean iguales volver a generar uno de los dos.
- 5.1. Se deberá comprobar que se han generado los cartones (ha entrado en la opción 1). En caso contrario, dar un mensaje de error y volver al menú principal sin hacer nada más.
- 5.2. Inicializar el array bombo a 0 con el procedimiento InicializarArray.
- 5.3. Generar un número para simular la bola con la función GenerarNombre.
- 5.4. Imprimir el número que ha salido y utilizar la función ImprimirBolesSortides para imprimir los números de las bolas que han salido a lo largo de la partida.
- 5.5. Utilizar la función Escrutinio para comprobar si alguno de los dos jugadores ha conseguido hacer Bingo.
- 5.6. En caso de que no haya ningún ganador volver al punto 5.3
6. Si la opción es 3: imprimir el marcador de las partidas que se han jugado y mostrar el mensaje generado por el procedimiento ImprimirComentari.
7. Si la opción es 4, imprimir el marcador final y mostrar el mensaje: "Saliendo del programa..."
8. Cualquier otra opción, escribir el mensaje: "Opción no permitida".
9. Repetir los pasos 3 a 9, hasta que la opción del menú escogida sea la 4.

Ejercicio 7 (0.5 puntos)
Declarar un nuevo tipo de datos, `TCarton`, como un registro con los campos: `Ident_Carton` (valor entero), `Numeros` (array de 25 enteros) y `Nombre_Jugador` (cadena de 50 caracteres).
Hacer un procedimiento llamado `Imprimir` que reciba como parámetro un registro del tipo `TCarton` e imprima el identificador del cartón y el nombre del jugador separandolos con el carácter `_`.

Ejercicio 8 (1 punto)
Hacer una función llamada `Es_MatPositiva` que reciba como parámetros: una matriz de enteros de 4 columnas, el número de filas de la matriz, y el número de columnas de la matriz, y devuelva 1 si todos los valores de la matriz son positivos o cero, y un 0 en caso contrario.

Ejercicio 9 (1 punto)
Escribir la función `Elevar` que devuelva el resultado de elevar un número al cuadrado. La función tendrá 2 parámetros de tipo `int`. El primero será la base y el segundo será el resultado de la operación en caso de que se pueda realizar. La función devolverá un 1 si se ha podido realizar la operación y un 0 en caso contrario.
NOTA: El valor de la base debe estar incluido en el intervalo [46.340,46.340] para evitar overflow.

Nom estudiant: Alícia Grado Serrà

Per la funció anomenada `Auxiliar` per saber si uns array tenen els mateixos valors menys la funció rebrà dos arrays i la seva dimensió. Retornarà true en el cas que siguin iguals i False en cas contrari.

NOTA: Podeu suposar que els dos arrays tenen la mateixa dimensió.

Important: Recordeu que cal donar les millors solucions possibles en cada exercici. A més de funcionar correctament, el codi ha d'estar ben programat (codi clar, amb les instruccions més adients, sense operacions ni variables innecessàries, etc.)

Les funcions i procediments de les preguntes de la 1 a la 6 formen part d'un únic programa i, per tant, ha d'haver coherència entre les seves definicions, i la forma en què s'utilitzen en altres preguntes. El context del programa a desenvolupar, el teniu a continuació:

El Bingo de 75 boles és un joc d'atzar on hi ha 75 boles numerades de l'1 al 75 dins del bombo. Els cartrons tenen un total de 25 nombres. En aquest tipus de bingo, es van treient boles del bombo i es guanya un premi quan han sortit tots els nombres del nostre cartó. Això es coneix com a bingo o full House.

Volem fer un programa que permeti jugar al Bingo a dos jugadors. Els jugadors podran jugar tantes partides com vulguin i caldrà portar el compte de quantes partides ha guanyat cadascú.

Exercici 1 (1 punt)

Fer un procediment anomenat `CrearCartó` per a generar un cartó. S'han de generar 25 nombres del 1 al 75. S'ha d'assegurar que hi hagin:

- 5 nombres del 1' al 15,
- 5 nombres del 16 al 30,
- 5 nombres del 31 al 45,
- 5 nombres del 46 al 60 i
- 5 nombres del 61 al 75.

El procediment rebrà un array (que serà el cartó) i la dimensió de l'array. Retornarà l'array ple amb els números ordenats de menor a major.

UTILITZEU (`no implementar`) els següents procediments o funcions:

- `int GenerarNúmero (int min, int max, int a[], int dim)`
Genera un nombre aleatori entre min i max, i revisa que no estigui a l'array a de dimensió dim.
- `void Ordena (int a[], int dim)`
Ordena l'array a de dimensió dim de menor a major.

$$70 + 70 = 140$$

$$140 - 3 = \underline{137}$$

$$80 + 120 = 200$$

$$200 \div 10 = 20$$

Resultat: 20

Exercici 3 (1 punt)

Fer un procediment anomenat `ImprimirBolesSortides` per imprimir els nombres de les boles que ja han sortit. El procediment rebrà un array i la seva dimensió. L'array contindrà informació sobre si un nombre ha sortit o no. Si el valor d'una determinada posició (`index`) és un 1, voldrà dir que la bola corresponent ha sortit i si el és un 0, voldrà dir que aquella bola no ha sortit.

NOTA: Recordeu que els números de les boles van de 1 a 75.

Fer una funció anomenat Escrutini per saber si un cartró ha aconseguit Bingo. La funció rebrà un array que serà el cartró, la dimensió del cartró i un array (bombo) que contingrà informació dels nombres que han sortit. En l'array bombo si el valor d'una determinada posició (index) és un 1, vindrà dir que la bola correspondent ha sortit i si el és un 0, vindrà dir que aquella bola no ha sortit.

La funció retornarà True en el cas que s'hagi aconseguit Bingo i False en cas contrari.

Suposeu que teniu els següents procediments i funcions ja implementats:

```
void Menuprincipal()
{
    cout << " --- Menu Principal --- " << endl;
    cout << "1. - Asignar cartrons " << endl;
    cout << "2. - Jugar" << endl;
    cout << "3.. Marcador" << endl;
    cout << "4.. Sortir" << endl;
}
```

Mostra per pantalla un menú amb les opcions del codi: 1. Assignar cartrons, 2. Jugar, etc.

```
void InicialitzarArray(int a[], int dim, int v)
{
    int Generarnombre(int min, int max, int a[], int dim)
    {
        // Retorna un enter aleatori entre min i max, controlant que no estigui a l'array a de dimensió dim
        // Inicialitza el vector a de dimensió dim al valor v
    }
}
```

int Generarnombre(int min, int max, int a[], int dim)
{
 // Retorna un enter aleatori entre min i max, controlant que no estigui a l'array a de dimensió dim
 // Inicialitza el vector a de dimensió dim al valor v
}

Fer un programa complet (declaracions globals i funció main()) que segueixi els següents passos:

1. Declarar (on corresponguï) les constants necessàries per tal que el programa sigui fàcilment modificable.
2. Declarar (on corresponguï) els arrays necessaris per representar els cartrons de dos jugadors cada jugador. En el cas que la diferència sigui:
 - menor o igual que 1, escriure el missatge "Està molt igualat".
 - entre 2 i 5, el missatge hauria de ser "La remuntada és possible".
 - Major o igual que 6, el missatge haurà de ser "Avui és el teu dia de sort".
3. Utilitzar el procediment Menuprincipal() i llegir l'opció escollida.
4. Si l'opció és 1, generar dos cartrons (un per a cada jugador) seguint els passos següents:
 - 4.1. Inicialitzar els cartrons dels jugadors tot a 0 amb el procediment InicialitzarArray.
 - 4.2. Utilitzar el procediment CrearCartró per crear els cartrons del jugador 1 i del jugador 2.
5. Utilitzar el procediment CrearCartró per crear els cartrons del jugador 1 i del jugador 2.
6. Comprovar si els dos cartrons són diferents utilitzant la funció ArraysIguals. En cas que siguin iguals tornar a generar un dels dos.
7. Si l'opció és 2, implementa el joc seguint els passos següents:
 - 7.1. S'haurà de comprovar que s'han generat els cartrons (s'ha entrat a l'opció 1). En cas contrari, donar un missatge d'error i tornar al menú principal sense fer res més.
 - 7.2. Inicialitzar l'array bombo a 0 amb el procediment InicialitzarArray.
 - 7.3. Generar un nombre per simular la bola amb la funció Generarnombre.
 - 7.4. Imprimir el nombre que ha sortit i utilitzar la funció ImprimirBolesSortides per imprimir els nombres de les boles que han sortit al llarg de la partida.
 - 7.5. Utilitzar la funció Escrutini per comprovar si algun dels dos jugadors ha aconseguit fer Bingo.
 - 7.6. En cas que no hi hagi cap guanyador tornar al punt 5.3
 - 7.7. Si l'opció és 3: imprimir el marcador de les partides que s'han jugat i mostrar el missatge generat pel procediment ImprimirComentari.
 - 7.8. Si l'opció és 4, imprimir el marcador final i mostrar el missatge: "Sortint del programa . . ."
 - 7.9. Qualsevol altra opció, escriure el missatge: "Opció no permesa".
 - 7.10. Repetir els passos 3 a 9, fins que l'opció del menú escollida sigui la 4.

Declarar un nou tipus de dades, `TCartro`, com un registre amb els camps: `Ident_Cartro` (valor ente), `Numeros` (array de 25 enters) i `Nom_Jugador` (cadena de 50 caràcters).

Fer un procediment anomenat `Imprimir` que rebi com a paràmetre un registre del tipus `TCartro` i imprimeixi l'identificador del cartó i el nom del jugador separant-los amb el caràcter ‘.

Escriviu la funció `Elevar` que retorna el resultat d'elevat un nombre al quadrat. La funció tindrà dos paràmetres de tipus int. El primer serà la base i el segon serà el resultat de l'operació en el cas que es pugui realizar. La funció retornarà un 1 si s'ha pogut realizar l'operació i un 0 en cas contrari.

NOTA: El valor de la base ha d'estar inclòs en l'interval [-46.340,46.340] per evitar overflow.

Exercici 8 (1 punt)

Fer una funció anomenada `Es_MatPositiva` que rebi com a paràmetres: una matriu d'enters de 4 columnes, el número de files de la matriu, i el número de columnes de la matriu. i retorni 1 si tots els valors de la matriu són positius o zero, i un 0 en cas contrari.

Per la funció anomenada `ARRAYS` que accepta un array i una dimensió. La funció rebrà dos arrays i la seva dimensió. Retornarà `true` en el cas que siguin iguals i `false` en cas contrari.

NOTA: Podeu suposar que els dos arrays tenen la mateixa dimensió.

Important: Recordeu que cal donar les millors solucions possibles en cada exercici. A més de funcionar correctament, el codi ha d'estar ben programat (codi clar, amb les instruccions més adients, sense operacions ni variables innecessàries, etc.)

Les funcions i procediments de les preguntes de la 1 a la 6 formen part d'un únic programa i, per tant, ha d'haver coherentia entre les seves definicions, i la forma en què s'utilitzen en altres preguntes. El context del programa a desenvolupar, el teniu a continuació:

El Bingo de 75 boles és un joc d'atzar on hi ha 75 boles numerades de l'1 al 75 dins del bombo. Els cartrons tenen un total de 25 nombres. En aquest tipus de bingo, es van treient boles del bombo i es guanya un premi quan han sortit tots els nombres del nostre cartró. Això es coneix com a bingo o full House.

Volem fer un programa que permeti jugar al Bingo a dos jugadors. Els jugadors podran jugar tantes partides com vulguin i caldrà portar el compte de quantes partides ha guanyat cadascú.

Exercici 1 (1 punt)

Fer un procediment anomenat `CrearCartró` per a generar un cartró. S'han de generar 25 nombres del 1 al 75. S'ha d'assegurar que hi hagin:

- 5 nombres del 11 al 15,
- 5 nombres del 46 al 60 i
- 5 nombres del 61 al 75.
- 5 nombres del 31 al 45,

El procediment rebrà un array (que serà el cartró) i la dimensió de l'array. Retornarà l'array ple amb els números ordenats de menor a major.

UTILITZEU (no implementar) els següents procediments o funcions:

- `int GenerarNombre (int min, int max, int a[], int dim)`
Genera un nombre aleatori entre min,max, i revisa que no estigui a l'array a de dimensió dim.
- `void Ordena (int a[], int dim)`
Ordena l'array a de dimensió dim de menor a major.

$$a(0) = 35 - 55 = 55$$

$$55 \div 5 = \boxed{11}$$

Exercici 3 (1 punt)

Fer un procediment anomenat `ImprimirBolesSortides` per imprimir els nombres de les boles que ja han sortit. El procediment rebrà un array i la seva dimensió. L'array contingirà informació sobre si un nombre ha sortit o no. Si el valor d'una determinada posició (`index`) és un 1, voldrà dir que la bola corresponent ha sortit i si el és un 0, voldrà dir que aquella bola no ha sortit.

NOTA: Recordeu que els números de les boles van de 1 a 75.

$$\begin{aligned} 6^3 &= 12 = 51 \\ 51 \times 20 &= 1020 \\ \text{Resultat} &= 1020 \end{aligned}$$

Fer una funció anònima. Escrutini per saber si un cartó ha aconseguit Bingo. La funció rebrà un array que sera el cartó, la dimensió del cartó i un array (bombo) que contindrà informació dels sortits que han sortit. En l'array bombo si el valor d'una determinada posició (índex) és un 1, voldrà dir que la bola correspondent ha sortit; si el és un 0, voldrà dir que aquella bola no ha sortit.

La funció retornarà true en el cas que s'hagi aconseguit Bingo i false en cas contrari.

Exercici 6 (2,5 punts)
Suposeu que teniu els següents procediments i funcions ja implementats:

```
void InicialitzarArray(int a[], int dim, int v)
{
    cout << "— Menu Principal ---" << endl;
    cout << "1.- Asignar cartons" << endl;
    cout << "2.- Jugar" << endl;
    cout << "3.- Marcador" << endl;
    cout << "4.- Sortir" << endl;
}
```

Mostra per pantalla un menú amb les options del codi: 1. Assignar cartons, 2. Jugar, etc.

```
void InicialitzarArray(int a[], int dim, int v)
{
    cout << "— Menú Principal ---" << endl;
    cout << "1.- Asignar cartons" << endl;
    cout << "2.- Jugar" << endl;
    cout << "3.- Marcador" << endl;
    cout << "4.- Sortir" << endl;
}
```

Inicialitza el vector a de dimensió dim al valor v

int GenerarNombre(int min, int max, int a[], int dim)

Retorna un enter aleatori entre min i max, controlant que no estigui a l'array a de dimensió dim.

Fer un programa complet (declaracions globals i funció main) que segueixi els següents passos:

1. Declarar (on correspongui) les constants necessàries per tal que el programa sigui fàcilment modificable.
2. Declarar (on correspongui) els arrays necessaris per representar els cartons de dos jugadors anomenats cartoPlayer1 i cartoPlayer2. Declarar també un array, anomenat bombo, per controlar quins nombres han sortit i quins no.
3. Utilitzar el procediment MenuPrincipal() i llegir l'opció escollida.
4. Si l'opció és 1, generar dos cartons (un per a cada jugador) seguint els passos següents:
 - 4.1. Inicialitzar els cartons dels jugadors tot a 0 amb el procediment InicialitzarArray.
 - 4.2. Utilitzar el procediment CrearCarto per crear els cartons del jugador 1 i del jugador 2.
 - 4.3. Comprovar si els dos cartons són diferents utilitzant la funció ArraysIguals. En cas que siguin iguals tornar a generar un dels dos.
5. Si l'opció és 2, implementa el joc seguint els passos següents:
 - 5.1. S'haurà de comprovar que s'han generat els cartons (s'ha entrat a l'opció 1). En cas contrari, donar un missatge d'error i tornar al menú principal sense fer res més.
 - 5.2. Inicialitzar l'array bombo a 0 amb el procediment InicialitzarArray.
 - 5.3. Generar un nombre per simular la bola amb la funció GenerarNombre.
 - 5.4. Imprimir el nombre que ha sortit i utilitzar la funció ImprimirBolesSortides per imprimir els nombres de les botes que han sortit al llarg de la partida.
 - 5.5. Utilitzar la funció Escrutini per comprovar si algun dels dos jugadors ha aconseguit fer Bingo.
 - 5.6. En cas que no hi hagi cap guanyador tornar al punt 5.3
 6. Si l'opció és 3: imprimir el marcador de les partides que s'han jugat i mostrar el missatge generat pel procediment ImprimirComentari.
 7. Si l'opció es 4, imprimir el marcador final i mostrar el missatge: "Sortint del programa..."
 8. Qualsevol altra opció, escriure el missatge: "Opció no permesa".
 9. Repetir els passos 3 a 9, fins que l'opció del menú escollida sigui la 4.

Exercici 7 (0,5 punts)
Declarar un nou tipus de dades, TCartro, com un registre amb els camps: Ident_Cartro (valor enter), Numeros (array de 25 enters) i Nom_Jugador (cadena de 50 caràcters).

Fer un procediment anomenat Imprimir que rebi com a paràmetre un registre del tipus TCartro i imprimeix l'identificador del cartró i el nom del jugador separant-los amb el caràcter ‘,’.

Exercici 8 (1 punt)
Escriure la funció Elevar que retorna el resultat d'elevar un nombre al quadrat. La funció tindrà dos paràmetres del tipus int. El primer serà la base i el segon serà el resultat de l'operació en el cas que es pugui realizar. La funció retornarà un 1 si s'ha pogut realitzar l'operació i un 0 en cas contrari.

NOTA: El valor de la base ha d'estar inclos en l'interval [-46,340;46,340] per evitar overflow.

Examen Parcial (23 de Novembre de 2020) Grup: **NIU: 4343829**
Nom estudiant: **Marcos Rodríguez Heras**

Reia l'unció anomenada **ALERTA** i que s'ha de fer servir si una situació excepcional es produeix.
La funció rebrà dos arrays i la seva dimensió. Retornarà **True** en el cas que siguin iguals i **False** en cas contrari.

NOTA: Podeu suposar que els dos arrays tenen la mateixa dimensió.

Important: Recordeu que cal donar les millors solucions possibles en cada exercici. A més de funcionar correctament, el codi ha d'estar ben programat (codi clar, amb les instruccions més adients, sense operacions ni variables innecessàries, etc.)

Les funcions i procediments de les preguntes de la 1 a la 6 formen part d'un únic programa i, per tant, ha d'haver coherentia entre les seves definicions, i la forma en què s'utilitzen en altres preguntes. El context del programa a desenvolupar, el teniu a continuació:

El Bingo de 75 boles és un joc d'atzar on hi ha 75 boles numerades de 1 al 75 dins del bumbo. Els cartrons tenen un total de 25 nombres. En aquests tipus de bingo, es van treient boles del bumbo i es guanya un premi quan han sortit tots els números del nostre cartó. Això es coneix com a bingo o full House.

Volem fer un programa que permeti jugar al Bingo a dos jugadors. Els jugadors podran jugar tantes partides com vulguin i caldrà portar el compte de quantes partides ha guanyat cadascú.

Exercici 1 (1 punt)

Fer un procediment anomenat **CrearCartro** per a generar un cartró. S'han de generar 25 nombres del 1 al 75. S'ha d'assegurar que hi hagin:

- 5 nombres del 1 al 15,
- 5 nombres del 16 al 30,
- 5 nombres del 46 al 60 i
- 5 nombres del 61 al 75.

El procediment rebrà un array (que serà el cartró) i la dimensió de l'array. Retornarà l'array ple amb els números ordenats de menor a major.

UTILITZEU (no implementar) els següents procediments o funcions:

- **int GeneraNombre (int min, int max, int a[], int dim)**
Genera un nombre aleatori entre min i max, i revisa que no estigui a l'array a de dimensió dim.
- **void Ordena (int a[], int dim)**
Ordena l'array a de dimensió dim de menor a major.

$$7^1 \times 3 = 21^3$$

$$21^3 + 328 = \underline{\underline{541}}$$

Exercici 3 (1 punt)

Fer un procediment anomenat **ImprimirBolesSortides** per imprimir els nombres de les boles que ja han sortit. El procediment rebrà un array i la seva dimensió. L'array contingrà informació sobre si un nombre ha sortit o no. Si el valor d'una determinada posició (**index**) és un 1, vindrà dir que la bola corresponent ha sortit i si el és un 0, vindrà dir que aquella bola no ha sortit.

NOTA. Recordeu que els números de les boles van de 1 a 75.

$$\begin{aligned} 90 - 8 &= 82 \\ 82 + 43 &= 125 \end{aligned}$$

Fer una funció anomenada Escrutini per saber si un cartó ha aconseguit Bingo. La funció rebrà un array que serà el cartó, la dimensió del cartó i un array (bombo) que contindrà informació dels números que han sortit. En l'array bombó si el valor d'una determinada posició (índex) és un 1, voldrà dir que la bola correspondent ha sortit i si el és un 0, voldrà dir que aquella bola no ha sortit.

La funció retornarà true en el cas que s'hagi aconseguit Bingo i false en cas contrari.

Suposet que teniu els següents procediments i funcions ja implementats:

```
Exercici 6 (2.5 punts)
void InicialitzarArray(int a[], int dim, int v)
{
    cout << "---- Menu Principal ---" << endl;
    cout << "1.- Asignar cartons " << endl;
    cout << "2.- Jugar" << endl;
    cout << "3.- Marcador" << endl;
    cout << "4.- Sortir" << endl;
}
```

Mostra per pantalla un menú amb les opcions del codi: 1. Asignar cartons, 2. Jugar, etc.

```
void InicialitzarArray(int a[], int dim, int v)
{
    //Initializa el vector a de dimensió dim al valor v
}
```

Retorna un enter aleatori entre min i max, controlant que no estigui a l'array a de dimensió dim.

Fer un programa complet (declaracions globals i funció main()) que segueixi els següents passos:

1. Declarar (on correspongui) les constants necessàries per tal que el programa sigui fàcilment modificable.
2. Declarar (on correspongui) els arrays necessaris per representar els cartons de dos jugadors anomenats cartoPlayer1 i cartoPlayer2. Declarar també un array, anomenat bombo, per controlar quins nombres han sortit i quins no.
3. Utilitzar el procediment MenuPrincipal() i llegir l'opció escollida.
4. Si l'opció és 1, generar dos cartons (un per a cada jugador) seguint els passos següents:
 - 4.1. Inicialitzar els cartons dels jugadors tot a 0 amb el procediment InicialitzarArray.
 - 4.2. Utilitzar el procediment CrearCarto per crear els cartons del jugador 1 i del jugador 2.
 - 4.3. Comprovar si els dos cartons són diferents utilitzant la funció ArraysIguals. En cas que siguin iguals tornar a generar un dels dos.
5. Si l'opció és 2, implementa el joc seguint els passos següents:
 - 5.1. S'haurà de comprovar que s'han generat els cartons (s'ha entrat a l'opció 1). En cas contrari, donar un missatge d'error i tornar al menú principal sense fer res més.
 - 5.2. Inicialitzar l'array bombo a 0 amb el procediment InicialitzarArray.
 - 5.3. Generar un nombre per simular la bola amb la funció GenerarNombre.
 - 5.4. Imprimir el nombre que ha sortit i utilitzar la funció ImprimirBolesSortides per imprimir els nombres de les botes que han sortit al llarg de la partida.
 - 5.5. Utilitzar la funció Escrutini per comprovar si algun dels dos jugadors ha aconseguit fer Bingo.
 - 5.6. En cas que no hi hagi cap guanyador tornar al punt 5.3
 6. Si l'opció és 3: imprimir el marcador de les partides que s'han jugat i mostrar el missatge generat pel procediment ImprimirComentari.
 7. Si l'opció és 4, imprimir el marcador final i mostrar el missatge: "Sortint del programa...."
 8. Qualsevol altra opció, escriure el missatge: "Opció no permetsa".
 9. Repetir els passos 3 a 9, fins que l'opció del menú escollida sigui la 4.

Declarar un nou tipus de dades, TCarro, com un registre amb els camps: Ident_Carro (valor enter), Numeros (array de 25 enters) i Nom_Jugador (cadena de 50 caràcters).

Fer un procediment anomenat Imprimir que rebi com a parametre un registre del tipus TCarro i imprimeix l'identificador del carro i el nom del jugador separant-los amb el caràcter ‘,’.

Exercici 9 (1 punt)
Escriure la funció Elevar que retorni el resultat d'elevat un nombre al quadrat. La funció tindrà paràmetres de tipus int. El primer serà la base i el segon serà el resultat de l'operació en el cas que es pugui realitzar. La funció retornarà un 1 si s'ha pogut realitzar l'operació i un 0 en cas contrari.

NOTA: El valor de la base ha d'estar inclos en l'interval [-46.340,46.340] per evitar overflow.

Exercici 8 (1 punt)

Fer una funció anomenada Es_MatPositiva que rebi com a paràmetres: una matriu d'enters de 4 columnes, el número de files de la matriu, i el número de columnes de la matriu, i retorna 1 si tots els valors de la matriu són positius o zero, i un 0 en cas contrari.

Examen Parcial (23 de Novembre de 2020)

Nom estudiant:

Grup:

443 444

NIU:

Fer la funció anomenada `arraysIguals` per saber si dos arrays tenen tots els valors iguals. La funció rebrà dos arrays i la seva dimensió. Retornarà `True` en el cas que siguin iguals i `False` en cas contrari.

NOTA: Podeu suposar que els dos arrays tenen la mateixa dimensió.

Important: Recordeu que cal donar les millors solucions possibles en cada exercici. A més de funcionar correctament, el codi ha d'estar ben programat (codi clar, amb les instruccions més adients, sense operacions ni variables innecessàries, etc.)

Les funcions i procediments de les preguntes de la 1 a la 6 formen part d'un únic programa i, per tant, ha d'haver coherentia entre les seves definicions, i la forma en què s'utilitzen en altres preguntes. El context del programa a desenvolupar, el teniu a continuació:

El Bingo de 75 boles és un joc d'atzar on hi ha 75 boles numerades de l'1 al 75 dins del bombo. Els cartells tenen un total de 25 nombres. En aquest tipus de bingo, es van treient boles del bombo i es guanya un premi quan han sortit tots els nombres del nostre cartell. Això es coneix com a bingo o full House.

Volem fer un programa que permeti jugar al Bingo a dos jugadors. Els jugadors podran jugar tantes partides com vulguin i caldrà portar el compte de quantes partides ha guanyat cada un.

Exercici 1 (1 punt)

Fer un procediment anomenat `CrearCartro` per a generar un cartell. S'han de generar 25 nombres del 1 al 75. S'ha d'assegurar que hi hagin:

- 5 nombres del 1^{er} al 15,
- 5 nombres del 16^{er} al 30,
- 5 nombres del 31^{er} al 45,
- 5 nombres del 46^{er} al 60 i
- 5 nombres del 61^{er} al 75.

El procediment rebrà un array (que serà el cartell) i la dimensió de l'array. Retornarà l'array ple amb els números ordenats de menor a major.

UTILITZEU (no implementar) els següents procediments o funcions:

- `int GenerarNombre (int min, int max, int a[], int dim)`
Genera un nombre aleatori entre min i max, i revisa que no estigui a l'array a de dimensió dim.
- `void Ordena (int a[], int dim)`
Ordene l'array a de dimensió dim de menor a major.

$$4 \times 5 = 20$$
$$20 + 4a = \underline{\underline{5a}}$$

Exercici 3 (1 punt)

Fer un procediment anomenat `ImprimirBolesSortides` per imprimir els nombres de les boles que ja han sortit. El procediment rebrà un array i la seva dimensió. L'array contingrà informació sobre si un nombre ha sortit o no. Si el valor d'una determinada posició (`index`) és un 1, vindrà dir que la bola corresponent ha sortit i si el és un 0, vindrà dir que aquella bola no ha sortit.

NOTA: Recordeu que els números de les boles van de 1 a 75.

$$8 \times 8 = 64$$
$$64 - 18 = 46$$

Resultat: 46

Fer una funció anomenada Escrutini per saber si un cartó ha aconseguit Bingo. La funció rebrà un array que serà el cartó, la dimensió del cartó i un array (bombo) que contindrà informació dels nombres que han sortit. En l'array bombó si el valor d'una determinada posició (index) és un 1, vindrà dir que la bola correspondent ha sortit i si el és un 0, vindrà dir que aquella bola no ha sortit.

La funció retornarà True en el cas que s'hagi aconseguit Bingo i False en cas contrari.

L'exercici té (25 punts) Suposet que tenim els següents procediments i funcions ja implementats:

```
void MenuPrincipal()
{
    cout << " --- Menu Principal --- " << endl;
    cout << "1. - Asignar cartons " << endl;
    cout << "2. - Jugar " << endl;
    cout << "3. - Marcador " << endl;
    cout << "4. - Sortir " << endl;
}
```

Mostra per pantalla un menú amb les opcions del codi: 1. Assignar cartons, 2. Jugar, etc.

```
void InicialitzarArray(int a[], int dim, int v)
Inicialitza el vector a de dimensió dim al valor v
```

int GenerarNombre(int min, int max, int a[], int dim)
Retorna un enter aleatori entre min i max, controlant que no estigui a l'array a de dimensió dim

Fer un programa complet (declaracions globals i funció main()) que segueixi els següents passos:

1. Declarar (on corresponguï) les constants necessàries per tal que el programa sigui fàcilment modificable.
2. Declarar (on corresponguï) els arrays necessaris per representar els cartons de dos jugadors anomenats cartroplayer1 i cartroplayer2. Declarar també un array, anomenat bombo, per controlar quins nombres han sortit i quins no.
3. Utilitzar el procediment MenuPrincipal() i llegir l'opció escollida.
4. Si l'opció es 1, generar dos cartons (un per a cada jugador) seguint els passos següents:
 - 4.1. Inicialitzar els cartons dels jugadors tot a 0 amb el procediment InicialitzarArray.
 - 4.2. Utilitzar el procediment CrearCartro per crear els cartons del jugador 1 i del jugador 2.
 - 4.3. Comprovar si els dos cartons són diferents utilitzant la funció ArraysIguals. En cas que siguin iguals tornar a generar un dels dos.
5. Si l'opció és 2, implementa el joc seguint els passos següents:
 - 5.1. S'haurà de comprovar que s'han generat els cartons (s'ha entrat a l'opció 1). En cas contrari, donar un missatge d'error i tornar al menú principal sense fer res més.
 - 5.2. Inicialitzar l'array bombo a 0 amb el procediment InicialitzarArray.
 - 5.3. Generar un nombre per simular la bola amb la funció GenerarNombre.
 - 5.4. Imprimir el nombre que ha sortit i utilitzar la funció ImprimirBolesSortides per imprimir els nombres de les botes que han sortit al llarg de la partida.
 - 5.5. Utilitzar la funció Escrutini per comprovar si algun dels dos jugadors ha aconseguit fer Bingo.
 - 5.6. En cas que no hi hagi cap guanyador tornar al punt 5.3
 6. Si l'opció és 3: imprimir el marcador de les partides que s'han jugat i mostrar el missatge generat pel procediment ImprimirComentari.
 7. Si l'opció és 4, imprimir el marcador final i mostrar el missatge: "Sortint del programa...".
 8. Qualsevol altra opció, escriure el missatge: "Opció no permesa".
 9. Repetir els passos 3 a 9, fins que l'opció del menú escollida sigui la 4.

Exercici 7 (0,5 punts)
Declarar un nou tipus de dades, TCartró, com un registre amb els camps: Ident_Cartró (valor enter), Numeros (array de 25 enters) i Nom_Jugador (cadena de 50 caràcters).

Fer un procediment anomenat Imprimir que rebi com a paràmetre un registre del tipus TCartró i imprimeix l'identificador del cartró i el nom del jugador separant-los amb el caràcter ‘_’.

Exercici 8 (1 punt)
Escriure la funció Elevar que retorna el resultat d'elevat un nombre al quadrat. La funció tindrà paràmetres de tipus int. El primer serà la base i el segon serà el resultat de l'operació en el cas que es pugui realizar. La funció retornarà un 1 si s'ha pogut realizar l'operació i un 0 en cas contrari.

NOTA: El valor de la base ha d'estar inclos en l'intervall [-46.340,46.340] per evitar overflow.

Important: Recordeu que cal donar les millors solucions possibles en cada exercici. A més de funcionar correctament, el codi ha d'estar ben programat (codi clar, amb les instruccions més adients, sense operacions ni variables innecessàries, etc.)

Les funcions i procediments de les preguntes de la 1 a la 6 formen part d'un únic programa i, per tant, ha d'haver coherència entre les seves definicions, i la forma en què s'utilitzen en altres preguntes. El context del programa a desenvolupar, el teniu a continuació:

El Bingo de 75 boles és un joc d'atzar on hi ha 75 boles numerades de l'1 al 75 dins del bumbo.

Els cartells tenen un total de 25 nombres. En aquest tipus de bingo, es van treient boles del bumbo i es guanya un premi quan han sortit tots els nombres del nostre cartell. Això es coneix com a bingo o full House.

Volem fer un programa que permeti jugar al Bingo a dos jugadors. Els jugadors podran jugar tantes partides com vulguin i caldrà portar el compte de quantes partides ha guanyat cadascú.

Exercici 1 (1 punt)

Fer un procediment anomenat `CrearCartell` per a generar un cartell. S'han de generar 25 nombres del 1 al 75. S'ha d'assegurar que hi hagin:

- 5 nombres del 11 al 15,
- 5 nombres del 46 al 60 i
- 5 nombres del 61 al 75.
- 5 nombres del 16 al 30,
- 5 nombres del 31 al 45,

El procediment rebrà un array (que serà el cartell) i la dimensió de l'array. Retornarà l'array ple amb els números ordenats de menor a major.

`UTILITZEU` (no implementar) els següents procediments o funcions:

- `int GeneraNombre (int min, int max, int a[], int dim)`
Genera un nombre aleatori entre min i max, i revisa que no estigui a l'array a de dimensió dim.
- `void Ordena (int a[], int dim)`
Ordena l'array a de dimensió dim de menor a major.

$$40 \times 20 = 800$$

$$\underline{50}$$

Fer la funció anomenada `ArraysIguals` per saber si dos arrays tenen tous els mateus elements. La funció rebrà dos arrays i la seva dimensió. Retorna `True` en el cas que siguin iguals i `False` en cas contrari.

NOTA: Podeu suposar que els dos arrays tenen la mateixa dimensió.

$$12 \times 2 = 24$$

$$80 + 15 = 95$$

Resultado :

Exercici 3 (1 punt)

Fer un procediment anomenat `ImprimirBolesSortides` per imprimir els nombres de les boles que ja han sortit. El procediment rebrà un array i la seva dimensió. L'array contingirà informació sobre si un nombre ha sortit o no. Si el valor d'una determinada posició (`index`) és un 1, voldrà dir que la bola corresponent ha sortit; si el és un 0, voldrà dir que aquella bola no ha sortit.

NOTA: Recordeu que els números de les boles van de 1 a 75.

Fer una funció anomenada Escrutini per saber si un cartó ha aconseguit Bingo. La funció rebrà un array que serà el cartó, la dimensió del cartó i un array (bombo) que contingrà informació dels nombres que han sortit. En l'array bombo si el valor d'una determinada posició (index) és un 1, voldrà dir que la bola correspondent ha sortit i si el és un 0, voldrà dir que aquella bola no ha sortit.

La funció retornarà True en el cas que s'hagi aconseguit Bingo i False en cas contrari.

Suposeu que teniu els següents procediments i funcions ja implementats:

```
Exercici 6 (2,5 punts)
void MenuPrincipal()
{
    cout << "--- Menu Principal ---" << endl;
    cout << "1.- Asignar cartones" << endl;
    cout << "2.- Jugar" << endl;
    cout << "3.- Marcador" << endl;
    cout << "4.- Salir" << endl;
}
```

Mostra per pantalla un menú amb les opcions del codi: 1. Asignar cartons. 2. Jugar, etc.

```
void InicializarArray(int a[], int dim, int v)
{
    int GenerarNombre(int min, int max, int a[], int dim);
    int i;
    for(i = 0; i < dim; i++)
        a[i] = GenerarNombre(min, max, a[i], dim);
}
```

Inicialitza el vector a de dimensió dim al valor v

Retorna un enter aleatori entre min i max, controlant que no estigui a l'array a de dimensió dim

Fer un programa complet (declaracions globals i funció main()) que segueixi els següents passos:

1. Declarar (o correspongui) les constants necessàries per tal que el programa sigui fàcilment modificable.
2. Declarar (o correspongui) els arrays necessaris per representar els cartons de dos jugadors anomenats cartoPlayer1 i cartoPlayer2. Declarar també un array, anomenat bombo, per controlar quins nombres han sortit i quins no.
3. Utilitzar el procediment MenuPrincipal() i llegir l'opció escollida.
4. Si l'opció és 1, generar dos cartons (un per a cada jugador) seguint els passos següents:
 - 4.1. Inicialitzar els cartons dels jugadors tot a 0 amb el procediment InicializarArray.
 - 4.2. Utilitzar el procediment CrearCarto per crear els cartons del jugador 1 i del jugador 2.
 - 4.3. Comprovar si els dos cartons són diferents utilitzant la funció ArraysIguals. En cas que siguin iguals tornar a generar un dels dos.
5. Si l'opció és 2, implementa el joc seguint els passos següents:
 - 5.1. Shaurà de comprovar que s'han generat els cartons (s'ha entrat a l'opció 1). En cas contrari, donar un missatge d'error i tornar al menú principal sense fer res més.
 - 5.2. Inicialitzar l'array bombo a 0 amb el procediment InicializarArray.
 - 5.3. Generar un nombre per simular la bola amb la funció GenerarNombre.
 - 5.4. Imprimir el nombre que ha sortit i utilitzar la funció ImprimirBolesSortides per imprimir els nombres de les boles que han sortit al llarg de la partida.
 - 5.5. Utilitzar la funció Escrutini per comprovar si algun dels dos jugadors ha aconseguit fer Bingo.
 - 5.6. En cas que no hi hagi cap guanyador tornar al punt 5.3
 6. Si l'opció és 3: imprimir el marcador de les partides que s'han jugat i mostrar el missatge generat pel procediment ImprimirComentari.
 7. Si l'opció és 4, imprimir el marcador final i mostrar el missatge: "Sortint del programa . . .".
 8. Qualsevol altra opció, escriure el missatge: "Opció no permesa".
 9. Repetir els passos 3 a 9, fins que l'opció del menú escollida sigui la 4.

Declarar un nou tipus de dades, TCarro, com un registre amb els camps: Ident_Carro (valor enter), Numeros (array de 25 enters) i Nom_Jugador (cadena de 50 caràcters).

Fer un procediment anomenat Imprimir que rebi com a paràmetre un registre del tipus TCarro i imprimeix l'identificador del carro i el nom del jugador separant-los amb el caràcter ‘:’.

Exercici 9 (1 punt)

Escriviu la funció Elevar que retorni el resultat d'elevar un nombre al quadrat. La funció tindrà dos paràmetres de tipus int. El primer serà la base i el segon serà el resultat de l'operació en el cas que es pugui realitzar. La funció retornarà un 1 si s'ha pogut realitzar l'operació i un 0 en cas contrari.

NOTA: El valor de la base ha d'estar inclos en l'intervall [-46.340,46.340] per evitar overflow.

Exercici 8 (1 punt)

Fer una funció anomenada Es_MatPositiva que rebi com a paràmetres: una matriu d'enters de 4 columnes, el numero de files de la matriu, i el número de columnes de la matriu, i retorna 1 si tots els valors de la matriu són positius o zero, i un 0 en cas contrari.

Important: Recordeu que cal donar les millors solucions possibles en cada exercici. A més de funcionar correctament, el codi ha d'estar ben programat (codi clar, amb les instruccions més adients, sense operacions ni variables innecessàries, etc.)

Les funcions i procediments de les preguntes de la 1 a la 6 formen part d'un únic programa i, per tant, ha d'haver coherència entre les seves definicions, i la forma en què s'utilitzen en altres preguntes. El context del programa a desenvolupar, el teniu a continuació:

El Bingo de 75 boles és un joc d'atzar on hi ha 75 boles numerades de l'1 al 75 dins del bombo. Els cartells tenen un total de 25 nombres. En aquest tipus de bingo, es van treient boles del bombo i es guanya un premi quan han sortit tots els números del nostre cartell. Això es coneix com a bingo o full House.

Volem fer un programa que permeti jugar al Bingo a dos jugadors. Els jugadors podran jugar tantes partides com vulguin i caldrà portar el compte de quantes partides ha guanyat cadascú.

Exercici 1 (1 punt)

Fer un procediment anomenat `CrearCartell` per a generar un cartell. S'han de generar 25 nombres del 1 al 75. S'ha d'assegurar que hi hagin:

- 5 nombres del 1 al 15,
- 5 nombres del 16 al 30,
- 5 nombres del 41 al 60 i
- 5 nombres del 61 al 75.

El procediment rebrà un array (que serà el cartell) i la dimensió de l'array. Retornarà l'array ple amb els números ordenats de menor a major.

UTILITZEU (no implementar) els següents procediments o funcions:

- `int GenerarNombre (int min, int max, int a[], int dim)`
Genera un nombre aleatori entre min i max, i revisa que no estigui a l'array a de dimensió dim.
- `void Ordena (int a[], int dim)`
Ordene l'array a de dimensió dim de menor a major.

$$100 \times 20 = \underline{\underline{2000}}$$

$$90 + 15 = \underline{\underline{105}}$$

$$12 + 10 = 114$$

$$100 - 10 = 90$$

Resultados: 90

Exercici 3 (1 punt)

Fer un procediment anomenat `ImprimirBolesSortides` per imprimir els nombres de les boles que ja han sortit. El procediment rebrà un array i la seva dimensió. L'array contingrà informació sobre si un nombre ha sortit o no. Si el valor d'una determinada posició (index) és un 1, voldrà dir que la bola correspondent ha sortit i si el és un 0, voldrà dir que aquella bola no ha sortit.

NOTA: Recordeu que els números de les boles van de 1 a 75.

rer la funció anomenada `MLT` que calculi la suma de dues matrícules en cas contrari.

NOTA: Podeu suposar que els dos arrays tenen la mateixa dimensió.

Fer una funció anomenada Escrutini per saber si un cartó ha aconseguit Bingo. La funció rebrà un array que serà el cartó, la dimensió del cartó i un array (bombo) que contingrà informació dels nombres que han sortit. En l'array bombó si el valor d'una determinada posició (index) és un 1, voldrà dir que la bola corresponent ha sortit i si el és un 0, voldrà dir que aquella bola no ha sortit.

La funció retornarà true en el cas que s'hagi aconseguit Bingo i false en cas contrari.

Suposu que teniu els següents procediments i funcions ja implementats:

```
void MenuPrincipal()
{
    cout << " --- Menu Principal --- " << endl;
    cout << "1.- Asignar cartons " << endl;
    cout << "2.- Jugar" << endl;
    cout << "3.- Marcador" << endl;
    cout << "4.- Sortir" << endl;
}
```

Mostra per pantalla un menú amb les opcions del codi: 1. Asignar cartons, 2. Jugar, etc.

```
void InicialitzarArray(int a[], int dim, int v)
```

Inicialitza el vector a de dimensió dim al valor v
int GenerarNombre (int min, int max, int a[], int dim)

Retorna un enter aleatori entre min i max, controlant que no estigui a l'array a de dimensió dim

Fer un programa complet (declaracions globals i funció main()) que segueixi els següents passos:

1. Declarar (on correspongui) les constants necessàries per tal que el programa sigui fàcilment modificable.
2. Declarar (on correspongui) els arrays necessaris per representar els cartons de dos jugadors anomenats cartoPlayer1 i cartoPlayer2. Declarar també un array, anomenat bombo, per controlar quins nombres han sortit i quins no.
3. Utilitzar el procediment MenuPrincipal () i llegir l'opció escollida.
4. Si l'opció és 1, generar dos cartons (un per a cada jugador) seguint els passos següents:
 - 4.1. Inicialitzar els cartons dels jugadors tot a 0 amb el procediment InicialitzarArray.
 - 4.2. Utilitzar el procediment CrearCarto per crear els cartons del jugador 1 i del jugador 2.
- 4.3. Comprovar si els dos cartons són diferents utilitzant la funció ArraysIguals. En cas que siguin iguals tornar a generar un dels dos.
5. Si l'opció és 2, implementa el joc seguint els passos següents:
 - 5.1. Shaurà de comprovar que s'han generat els cartons (s'ha entrat a l'opció 1). En cas contrari, donar un missatge d'error i tornar al menú principal sense fer res més.
 - 5.2. Inicialitzar l'array bombo a 0 amb el procediment InicialitzarArray.
 - 5.3. Generar un nombre per simular la bola amb la funció GenerarNombre.
 - 5.4. Imprimir el nombre que ha sortit i utilitzar la funció ImprimirBolesSortides per imprimir els nombres de les boles que han sortit al llarg de la partida.
 - 5.5. Utilitzar la funció Escrutini per comprovar si algun dels dos jugadors ha aconseguit fer Bingo.
 - 5.6. En cas que no hi hagi cap guanyador tornar al punt 5.3
 6. Si l'opció és 3: imprimir el marcador de les partides que s'han jugat i mostrar el missatge generat pel procediment ImprimirComentari.
 7. Si l'opció és 4, imprimir el marcador final i mostrar el missatge: "Sortint del programa..."
 8. Qualsevol altra opció, escriure el missatge: "Opció no permesa".
 9. Repetir els passos 3 a 9, fins que l'opció del menú escollida sigui la 4.

Exercici 6 (2,5 punts)

Suposu que teniu els següents procediments i funcions ja implementats:

```
void MenuPrincipal()
{
    cout << " --- Menu Principal --- " << endl;
    cout << "1.- Asignar cartons " << endl;
    cout << "2.- Jugar" << endl;
    cout << "3.- Marcador" << endl;
    cout << "4.- Sortir" << endl;
}
```


Declarar un nou tipus de dades, TCartro, com un registre amb els camps: Ident_Cartro (valor ente), Numeros (array de 25 enters) i Nom_Jugador (cadena de 50 caràcters).

Fer un procediment anomenat Imprimir que rebi com a paràmetre un registre del tipus TCartro i imprimeixi l'identificador del cartró el nom del jugador separant-los amb el caràcter ','.

Exercici 7 (0,5 punts)

Escriviu la funció Elevar que retorni el resultat d'elevat un nombre al quadrat. La funció tindrà paràmetres de tipus int. El primer serà la base i el segon serà el resultat de l'operació en el cas que es pugui realitzar. La funció retornara un 1 si s'ha pogut realitzar l'operació i un 0 en cas contrari.

NOTA: El valor de la base ha d'estar inclòs en l'intervall [-46.340,46.340] per evitar overflow.

Exercici 8 (1 punt)

Fer una funció anomenada Es_MatPositiva que rebí com a paràmetres: una matriu d'enters de 4 columnes, el número de files de la matriu, i el número de columnes de la matriu. i retorna 1 si tots els valors de la matriu són positius o zero, i un 0 en cas contrari.

Examen Parcial (23 de Novembre de 2020) Grup:
Nom estudiant: Stan TRONCOSO GUERRAS NIU: 1249176

Important: Recordeu que cal donar les millors solucions possibles en cada exercici. A més de funcionar correctament, el codi ha d'estar ben programat (codi clar, amb les instruccions més adients, sense operacions ni variables innecessàries, etc.)

Les funcions i procediments de les preguntes de la 1 a la 6 formen part d'un únic programa i, per tant, ha d'haver coherentia entre les seves definicions, i la forma en què s'utilitzen en altres preguntes. El context del programa a desenvolupar, el teniu a continuació:

El Bingo de 75 boles és un joc d'atzar on hi ha 75 boles numerades de l'1 al 75 dins del bumbo.

Els cartons tenen un total de 25 nombres. En aquest tipus de bingo, es van treient boles del bumbo i es guanya un premi quan han sortit tots els nombres del nostre cartó. Això es coneix com a bingo o full House.

Volem fer un programa que permeti jugar al Bingo a dos jugadors. Els jugadors podran jugar tantes partides com vulguin i caldrà portar el compte de quantes partides ha guanyyat cadascú.

Exercici 1 (1 punt)

Fer un procediment anomenat CrearCartro per a generar un cartó. S'han de generar 25 nombres del 1 al 75. S'ha d'assegurar que hi hagin:

- 5 nombres del 11 al 15,
- 5 nombres del 16 al 30,
- 5 nombres del 46 al 60 i
- 5 nombres del 61 al 75.

Els carttons tenen un total de 25 nombres. En aquest tipus de bingo, es van treient boles del bumbo i es guanya un premi quan han sortit tots els nombres del nostre cartó. Això es coneix com a bingo o full House.

UTILITZEU (no implementar) els següents procediments o funcions:

- int GenerarNombre (int min, int max, int a[], int dim)
Genera un nombre aleatori entre min,max, i revisa que no estigui a l'array a de dimensió dim.
- void Ordena (int a[], int dim)
Ordena l'array a de dimensió dim de menor a major.

$$20 + 40 = \underline{60}$$

$$30 \times 2 = \underline{60}$$

Per la funció anomenada `AltaYSortida` ha de saure si una cosa és certa o no. La funció rebrà dos arrays i la seva dimensió. Retorna `True` en el cas que siguin iguals i `False` en cas contrari.

NOTA: Podeu suposar que els dos arrays tenen la mateixa dimensió.

$$100 + 83 = 183$$
$$21 \times 15 = 315$$

Resultat : 315

Exercici 3 (1 punt)

Fer un procediment anomenat ImprimirBolesSortides per imprimir els nombres de les boles que ja han sortit. El procediment rebrà un array i la seva dimensió. L'array contingdrà informació sobre si un nombre ha sortit o no. Si el valor d'una determinada posició (`index`) es un 1, voldrà dir que la bola correspondent ha sortit i si el és un 0, voldrà dir que aquella bola no ha sortit.

NOTA: Recordeu que els números de les boles van de 1 a 75.

Fer una funció anomenada `Escrutini` per saber si un cartó ha aconseguit Bingo. La funció rebrà un array que serà el cartó, la dimensió del cartó i un array (bombo) que contingà informació dels nombres que han sortit. En l'array bombó si el valor d'una determinada posició (`index`) és un 1, voldrà dir que la bola correspondent ha sortit i si el és un 0, voldrà dir que aquella bola no ha sortit.

La funció retornarà `True` en el cas que s'hagi aconseguit Bingo i `False` en cas contrari.

Suposeu que teniu els següents procediments i funcions ja implementats:

```

Exercici 6 (2,5 punts)
{
    void InicialitzarArray(int a[], int dim, int v)
    {
        cout << " --- Menu Principal --- " << endl;
        cout << "1.- Asignar cartons " << endl;
        cout << "2.- Jugar" << endl;
        cout << "3.- Marcador" << endl;
        cout << "4.- Sortir" << endl;
    }
}
```

Mostra per pantalla un menú amb les opcions del codi: 1. Asignar cartons, 2. Jugar, etc.

```
void InicialitzarArray(int a[], int dim, int v)
{
    cout << " --- Menu Principal --- " << endl;
    cout << "1.- Asignar cartons " << endl;
    cout << "2.- Jugar" << endl;
    cout << "3.- Marcador" << endl;
    cout << "4.- Sortir" << endl;
}
```

Retorna un enter aleatori entre `min` i `max`, controlant que no estigui a l'array `a` de dimensió `dim`

Fer un programa complet (declaracions globals i funció `main()`) que segueixi els següents passos:

1. Declarar (on correspongui) les constants necessàries per tal que el programa sigui fàcilment modificable.

2. Declarar (on correspongui) els arrays necessaris per representar els cartons de dos jugadors anomenats `cartoPlayer1` i `cartoPlayer2`. Declarar també un array, anomenat `bombo`, per controlar quins nombres han sortit i quins no.

3. Utilitzar el procediment `MenuPrincipal()` i llegir l'opció escollida.

4. Si l'opció és 1, generar dos cartons (un per a cada jugador) seguint els passos següents:

4.1. Inicialitzar els cartons dels jugadors tot a 0 amb el procediment `InicialitzarCarto`.

4.2. Utilitzar el procediment `CrearCarto` per crear els cartons del jugador 1 i del jugador 2.

4.3. Comprovar si els dos cartons són diferents utilitzant la funció `ArraysIguals`. En cas que siguin iguals tornar a generar un dels dos.

5. Si l'opció és 2, implementar el joc seguint els passos següents:

5.1. S'haurà de comprovar que s'han generat els cartons (`s'ha entrat a l'opció 1`). En cas contrari, donar un missatge d'error i tornar al menú principal sense fer res més.

5.2. Inicialitzar l'array `bombo` a 0 amb el procediment `InicialitzarBombo`.

5.3. Generar un nombre per simular la bola amb la funció `GenerarNombre`.

5.4. Imprimir el nombre que ha sortit i utilitzar la funció `ImprimirBolesSortides` per imprimir els nombres de les boles que han sortit al llarg de la partida.

5.5. Utilitzar la funció `Escrutini` per comprovar si algun dels dos jugadors ha aconseguit fer Bingo.

5.6. En cas que no hi hagi cap guanyador tornar al punt 5.3

6. Si l'opció és 3: imprimir el marcador de les partides que s'han jugat i mostrar el missatge general pel procediment `ImprimirComentari`.

7. Si l'opció és 4, imprimir el marcador final i mostrar el missatge: "Sortint del programa..."

8. Qualsevol altra opció, escriure el missatge: "Opció no permesa".

9. Repetir els passos 3 a 9, fins que l'opció del menú escollida sigui la 4.

Declarar un nou tipus de dades, TCarro, com un registre amb els camps: Ident_Carro (valor enter), Numeros (array de 25 enters) i Nom_Jugador (cadena de 50 caràcters).
Fer un procediment anomenat Imprimir que rebi com a parametre un registre del tipus TCarro i imprimeix l'identificador del carro i el nom del jugador separant-los amb el caràcter ','.

Exercici 9 (1 punt)
Escriviu la funció Elevar que retorni el resultat d'elevat un nombre al quadrat. La funció tindrà dos paràmetres de tipus int. El primer serà la base i el segon serà el resultat de l'operació en el cas que es pugui realizar. La funció retornarà un 1 si s'ha pogut realitzar l'operació i un 0 en cas contrari.

NOTA: El valor de la base ha d'estar inclos en l'interval [-46.340, 46.340] per evitar overflow.

Exercici 8 (1 punt)

Fer una funció anomenada Es_MatPositiva que rebi com a paràmetres: una matriu d'enters de 4 columnes, el número de files de la matriu, i el número de columnes de la matriu, i retorna 1 si tots els valors de la matriu són positius o zero, i un 0 en cas contrari.

Fer la funció anomenada `ArraysIguals` per saber si dos arrays tenen tots els valors idèntics. La funció rebrà dos arrays i la seva dimensió. Retornarà `true` en el cas que siguin iguals i `false` en cas contrari.

NOTA:

Podeu suposar que els dos arrays tenen la mateixa dimensió.

Important: Recordeu que cal donar les millors solucions possibles en cada exercici. A més de funcionar correctament, el codi ha d'estar ben programat (codi clar, amb les instruccions més adients, sense operacions ni variables innecessàries, etc.)

Les funcions i procediments de les preguntes de la 1 a la 6 formen part d'un únic programa i, per tant, ha d'haver coherència entre les seves definicions, i la forma en què s'utilitzen en altres preguntes. El context del programa a desenvolupar, el teniu a continuació:

El Bingo de 75 boles és un joc d'atzar on hi ha 75 boles numerades de l'1 al 75 dins del bumbo. Els cartons tenen un total de 25 nombres. En aquest tipus de bingo, es van treient boles del bumbo i es guanya un premi quan han sortit tots els nombres del nostre cartó. Això es coneix com a bingo o full House.

Volem fer un programa que permeti jugar al Bingo a dos jugadors. Els jugadors podran jugar tantes partides com vulguin i caixerà portar el compte de quantes partides ha guanyat cadascú.

Exercici 1 (1 punt)

Fer un procediment anomenat `CrearCartro` per a generar un cartó. S'han de generar 25 nombres del 1 al 75. S'ha d'assegurar que hi hagin:

- 5 nombres del 1 al 15. • 5 nombres del 46 al 60 !
- 5 nombres del 16 al 30. • 5 nombres del 61 al 75.
- 5 nombres del 31 al 45.

El procediment rebrà un array (que serà el cartó) i la dimensió de l'array. Retornarà l'array ple amb els números ordenats de menor a major.

UTILITZEU (no implementar) els següents procediments o funcions:

- `int GeneraNombre (int min, int max, int a[], int dim)`
Genera un nombre aleatori entre `min,max`, i revisa que no estigui a l'array a de dimensió `dim`.
- `void Ordena (int a[], int dim)`
Ordene l'array a de dimensió `dim` de menor a major.

$$22 \times 53 = 423$$

$$423 - 7 = \underline{\underline{121}}$$

Fer una funció anomenada Escrutini per saber si un cartó ha aconseguit Bingo. La funció rebrà un array que serà el cartó, la dimensió del cartó i un array (bombo) que contingrà informació dels nombres que han sortit. En l'array bombo si el valor d'una determinada posició (index) és un 1, vindrà dir que la bola corresponent ha sortit i si el és un 0, vindrà dir que a aquella bola no ha sortit.

La funció retornarà True en el cas que s'hagi aconseguit Bingo i False en cas contrari.

Exercici 4 (1 punt)
Suposeu que tenui els següents procediments i funcions ja implementats:

```
void MenuPrincipal()
{
    cout << " --- Menu Principal --- " << endl;
    cout << "1. - Asignar cartons " << endl;
    cout << "2. - Jugar" << endl;
    cout << "3. - Marcador" << endl;
    cout << "4. - Sortir" << endl;
}
```

Mostra per pantalla un menú amb les opcions del codi: 1. Asignar cartons. 2. Jugar, etc.

```
void InicialitzarArray(int a[], int dim, int v)
{
    int GenerarNombre(int min, int max, int a[], int dim)
    {
        // Implementació de GenerarNombre
    }
}
```

Inicialitza el vector a de dimensió dim al valor v
Retorna un enter aleatori entre min i max, controlant que no estigui a l'array a de dimensió dim

Fer un programa complet (declaracions globals i funció main()) que segueixi els següents passos:

1. Declarar (on correspongui) les constants necessàries per tal que el programa sigui fàcilment modificable.
2. Declarar (on correspongui) els arrays necessaris per representar els cartons de dos jugadors anomenats cartoPlayer1 i cartoPlayer2. Declarar també un array, anomenat bombo, per controlar quins nombres han sortit i quins no.
3. Utilitzar el procediment MenuPrincipal() i llegir l'opció escollida.
4. Si l'opció és 1, generar dos cartons (un per a cada jugador) seguint els passos següents:
 - 4.1. Inicialitzar els cartons dels jugadors tot a 0 amb el procediment InicialitzarArray.
 - 4.2. Utilitzar el procediment CrearCarto per crear els cartons del jugador 1 i del jugador 2.
 - 4.3. Comprovar si els dos cartons són diferents utilitzant la funció ArraysIguals. En cas que siguin iguals tornar a generar un dels dos.
5. Si l'opció és 2, implementa el joc seguint els passos següents:
 - 5.1. S'haurà de comprovar que s'han generat els cartons (s'ha entrat a l'opció 1). En cas contrari, donar un missatge d'error i tornar al menú principal sense fer res més.
 - 5.2. Inicialitzar l'array bombo a 0 amb el procediment InicialitzarArray.
 - 5.3. Generar un nombre per simular la bola amb la funció GenerarNombre.
 - 5.4. Imprimir el nombre que ha sortit i utilitzar la funció ImprimirBolesSortides per imprimir els nombres de les boles que han sortit al llarg de la partida.
 - 5.5. Utilitzar la funció Escrutini per comprovar si algun dels dos jugadors ha aconseguit fer Bingo.
 - 5.6. En cas que no hi hagi cap guanyador tornar al punt 5.3
 6. Si l'opció és 3, imprimir el marcador de les partides que s'han jugat i mostrar el missatge generat pel procediment ImprimirComentari.
 7. Si l'opció és 4, imprimir el marcador final i mostrar el missatge: "Sortint del programa..."
 8. Qualsevol altra opció, escriure el missatge: "Opció no permesa".
 9. Repetir els passos 3 a 9, fins que l'opció del menú escollida sigui la 4.

Exercici 7 (0,5 punts)

Declarar un nou tipus de dades, `TCartro`, com un registre amb els camps: `Ident_Cartro` (valor enter), `Numeros` (array de 25 enters) i `Nom_Jugador` (cadena de 50 caràcters).

Fer un procediment anomenat `Imprimir` que rebi com a paràmetre un registre del tipus `TCartro` i imprimeixi l'identificador del cartó i el nom del jugador separant-los amb el caràcter '-'.

Exercici 9 (1 punt)

Escriure la funció `Elevar` que retorna el resultat d'elevat un nombre al quadrat. La funció tindrà paràmetres de tipus `int`. El primer serà la base i el segon serà el resultat de l'operació en el cas que es pugui realizar. La funció retornarà un 1 si s'ha pogut realitzar l'operació i un 0 en cas contrari.

NOTA: El valor de la base ha d'estar inclos en l'intervall [-46.340,46.340] per evitar overflow.

Exercici 8 (1 punt)

Fer una funció anomenada `E_S_MatPositiva` que rebi com a paràmetres: una matriu d'enters de 4 columnes, el número de files de la matriu, i el número de columnes de la matriu, i retorna 1 si tots els valors de la matriu són positius o zero, i un 0 en cas contrari.

Nom estudiant: Javier Falcone Locke

Important: Recordeu que cal donar les millors solucions possibles en cada exercici. A més de funcionar correctament, el codi ha d'estar ben programat (codi clar, amb les instruccions més adients, sense operacions ni variables innecessàries, etc.) Les funcions i procediments de les preguntes de la 1 a la 6 formen part d'un únic programa i, per tant, ha d'haver coherentia entre les seves definicions, i la forma en què s'utilitzen en altres preguntes. El context del programa a desenvolupar, el teniu a continuació:

El Bingo de 75 boles és un joc d'atzar on hi ha 75 boles numerades de l'1 al 75 dins del bombó. Els cartrons tenen un total de 25 nombres. En aquest tipus de bingo, es van treient boles del bombó i es guanya un premi quan han sortit tots els nombres del nostre cartó. Això es coneix com a bingo o full House.

Volem fer un programa que permeti jugar al Bingo a dos jugadors. Els jugadors podran jugar tantes partides com vulguin i caldrà portar el compte de quantes partides ha guanyat cadascú.

Exercici 1 (1 punt)

Fer un procediment anomenat `CrearCartró` per a generar un cartró. S'han de generar 25 nombres del 1 al 75. S'ha de segregar que hi hagin:

- 5 nombres del 11 al 15.
- 5 nombres del 46 al 60 i
- 5 nombres del 61 al 75.
- 5 nombres del 16 al 30.
- 5 nombres del 31 al 45.

El procediment rebrà un array (que serà el cartró) i la dimensió de l'array. Retornarà l'array ple amb els números ordenats de menor a major.

UTILITZEU (no implementar) els següents procediments o funcions:

- `int GenerarNombre (int min, int max, int a[], int dim)`
Genera un nombre aleatori entre min i max, i revisa que no estigui a l'array a de dimensió dim.
- `void Ordena (int a[], int dim)`
Ordena l'array a de dimensió dim de menor a major.

Exercici 2 (1 punt)
Fer la funció anomenada `ArraysIguals` per saber si dos arrays tenen tots els valors idèntics. La funció rebrà dos arrays i la seva dimensió. Retornarà True en el cas que siguin iguals i False en cas contrari.

NOTA: Podeu suposar que els dos arrays tenen la mateixa dimensió.

$$4 \times 3 = 12$$

$$12 \times 2 = 36$$

Resolt: 36

Exercici 3 (1 punt)

Fer un procediment anomenat `ImprimirBolesSortides` per imprimir els nombres de les boles que ja han sortit. El procediment rebrà un array i la seva dimensió. L'array contingrà informació sobre si un nombre ha sortit o no. Si el valor d'una determinada posició (index) és un 1, voldrà dir que la bola corresponent ha sortit i si el és un 0, voldrà dir que aquella bola no ha sortit.

NOTA: Recordeu que els números de les boles van de 1 a 75.

Exercici 4 (1 punt)
Fer una funció anomenada Escrutini per saber si un cartró ha aconseguit Bingo. La funció rebrà un array que serà el cartró, la dimensió del cartró i un array (bombo) que contindrà informació dels nombres que han sortit. En l'array bombo si el valor d'una determinada posició (index) és un 1, voldrà dir que la bola corresponent ha sortit i si el és un 0, voldrà dir que aquella bola no ha sortit.

La funció retornarà true en el cas que s'hagi aconseguit Bingo i false en cas contrari.

```
int Escrutini(int *cart, int tam, int *bombo)
{
    int i;
    for(i=0; i<tam; i++)
    {
        if(cart[i] == 1 && bombo[i] == 1)
            return true;
    }
    return false;
}
```

Exercici 6 (2,5 punts)
Suposeu que teniu els següents procediments i funcions ja implementats:

```
void MenuPrincipal()
{
    cout << "---- Menu Principal ----" << endl;
    cout << "1.- Asignar cartrons" << endl;
    cout << "2.- Jugar" << endl;
    cout << "3.- Marcador" << endl;
    cout << "4.- Sortir" << endl;
}
```

Mostra per pantalla un menú amb les opcions del codi: 1. Asignar cartrons, 2. Jugar, etc.

```
void InicialitzarArray(int a[], int dim, int v)
{
    for(int i=0; i<dim; i++)
        a[i] = v;
}
```

Retorna un enter aleatori entre min i max, controlant que no estigui a l'array a de dimensió dim

```
int GenerarNombre (int min, int max, int a[], int dim)
```

Inicialitza el vector a de dimensió dim al valor v

Fer un procediment anomenat ImprimirComentari per imprimir un comentari sobre el marcador. El procediment rebrà dos valors enters que seran el nombre de partides guanyades per cada jugador. En el cas que la diferència sigui:

- menor o igual que 1, escriure el missatge "Està molt igualat".
- entre 2 i 5, el missatge hauria de ser "La remuntada és possible".
- Major o igual que 6, el missatge haurà de ser "Avui és el teu dia de sort".

NOTA: podeu utilitzar la funció abs() per obtenir el valor absolut d'un nombre.

Exercici 5 (1 punt)
Fer un procediment anomenat ImprimirComentari per imprimir un comentari sobre el marcador. El procediment rebrà dos valors enters que seran el nombre de partides guanyades per cada jugador. En el cas que la diferència sigui:

- menor o igual que 1, escriure el missatge "Està molt igualat".
- entre 2 i 5, el missatge hauria de ser "La remuntada és possible".
- Major o igual que 6, el missatge haurà de ser "Avui és el teu dia de sort".
- Utilitzar el procediment MenuPrincipal() i llegir l'opció escollida.
- Si l'opció és 1, generar dos cartrons (un per a cada jugador) seguint els passos següents:
 - Declarar (on corresponguí) les constants necessàries per tal que el programa sigui fàcilment modificable.
 - Declarar (on corresponguí) els arrays necessaris per representar els cartrons de dos jugadors anomenats cartoCoplayer1 i cartoCoplayer2. Declarar també un array, anomenat bombo, per controlar quins nombres han sortit quins no.
 - Utilitzar el procediment MenuPrincipal() i llegir l'opció escollida.
 - Si l'opció és 1, generar dos cartrons (un per a cada jugador) seguint els passos següents:
 - Inicialitzar els cartrons dels jugadors tot a 0 amb el procediment InicialitzarArray.
 - Utilitzar el procediment CreaCartro per crear els cartrons del jugador 1 i del jugador 2.
 - Comprovar si els dos cartrons son diferents utilitzant la funció ArraysIguals. En cas que siguin iguals tornar a generar un dels dos.
 - Si l'opció és 2, implementa el joc següent els passos següents:
 - S'haura de comprovar que s'han generat els cartrons (s'ha entrat a l'opció 1). En cas contrari, donar un missatge d'error i tornar al menú principal sense fer res més.
 - Inicialitzar l'array bombo a 0 amb el procediment InicialitzarArray.
 - Generar un nombre per simular la bola amb la funció GenerarNombre.
 - Imprimir el nombre que ha sortit i utilitzar la funció ImprimirBolesSortides per imprimir els nombres de les boles que han sortit al llarg de la partida.
 - Utilitzar la funció Escrutini per comprovar si algun dels dos jugadors ha aconseguit fer Bingo.
 - En cas que no hi hagi cap guanyador tornar al punt 5.3
 - Si l'opció és 3: imprimir el marcador de les partides que s'han jugat i mostrar el missatge generat pel procediment ImprimirComentari.
 - Si l'opció és 4, imprimir el marcador final i mostrar el missatge: "Sortint del programa...".
 - Qualsevol altra opció, escriure el missatge: "Opció no permesa".
 - Repetir els passos 3 a 9 fins que l'opció del menú escollida sigula 4.

Exercici 7 (0.5 punts)
Declarar un nou tipus de dades, TCartro, com un registre amb els camps: Ident_Cartro (valor enter), Numeros (array de 25 enters) i Nom_Jugador (cadena de 50 caràcters).

Fer un procediment anomenat Imprimir que rebi com a paràmetre un registre del tipus TCartro i imprimeix l'identificador del cartró i el nom del jugador separant -los amb el caràcter ','.

Exercici 9 (1 punt)
Escrivre la funció Elevar que retorni el resultat de elevar un nombre al quadrat. La funció tindrà dos paràmetres de tipus int. El primer serà la base i el segon serà el resultat de l'operació en el cas que es pugui realitzar. La funció retornarà un 1 si s'ha pogut realitzar l'operació i un 0 en cas contrari.

NOTA: El valor de la base ha d'estar inclos en l'intervall [-46.340,46.340] per evitar overflow.

Exercici 8 (1 punt)

Fer una funció anomenada Es_MatPositiva que rebi com a paràmetres: una matriu d'enters de 4 columnes, el número de files de la matriu, i el número de columnes de la matriu, i retorna 1 si tots els valors de la matriu són positius o zero, i un 0 en cas contrari.

Examen Parcial (23 de Novembre de 2020)

Nom estudiant: Anna Korffs Stewart **Grup:** 43 **NIU:** 3509069

Important: Recordeu que cal donar les millors solucions possibles en cada exercici. A més de funcionar correctament, el codí ha d'estar ben programat (codi clar, amb les instruccions més adients, sense operacions ni variables innecessàries, etc.)

Les funcions i procediments de les preguntes de la 1 a la 6 formen part d'un únic programa i, per tant, ha d'haver coherentia entre les seves definicions, i la forma en què s'utilitzen en altres preguntes. El context del programa a desenvolupar, el teniu a continuació:

El Bingo de 75 boles és un joc d'atzar on hi ha 75 boles numerades de l'1 al 75 dins del bombo. Els cartells tenen un total de 25 nombres. En aquests tipus de bingo, es van treient boles del bombo i es guanya un premi quan han sortit tots els nombres del nostre cartell. Això es coneix com a bingo o full House.

Volem fer un programa que permeti jugar al Bingo a dos jugadors. Els jugadors podran jugar tantes partides com vulguin i caldrà portar el compte de quantes partides ha guanyat cadaescú.

Exercici 1 (1 punt)
Fer un procediment anomenat CrearCartell per a generar un cartell. S'han de generar 25 nombres del 1 al 75. S'ha d'assegurar que hi hagin:

- 5 nombres del 1^{er} al 15.
- 5 nombres del 16 al 30.
- 5 nombres del 31 al 45.
- 5 nombres del 46 al 60 i
- 5 nombres del 61 al 75.

El procediment rebrà un array (que serà el cartell) i la dimensió de l'array. Retornarà l'array ple amb els números ordenats de menor a major.

UTILITZEU (no implementar) els següents procediments o funcions:

- int GenerarNombre (int min, int max, int a[], int dim)
Genera un nombre aleatori entre min i max, i revisa que no estigui a l'array a de dimensió dim.
- void Ordena (int a[], int dim)
Ordene l'array a de dimensió dim de menor a major.

$$20 \times 10 = 200$$
$$200 + 50 = \underline{\underline{250}}$$

Per la funció anomenada `ARRAYS_IGUALS` per saber si dues arrays tenen els mateixos valors menys en cas contrari.

NOTA: Podeu suposar que els dos arrays tenen la mateixa dimensió.

Fer un procediment anomenat `ImprimirBolesSortides` per imprimir els nombres de les boles que ja han sortit. El procediment rebrà un array i la seva dimensió. L'array contindrà informació sobre si un nombre ha sortit o no. Si el valor d'una determinada posició (`index`) és un 1, voldrà dir que la bola corresponent ha sortit i si el és un 0, voldrà dir que aquella bola no ha sortit.

NOTA: Recordeu que els números de les boles van de 1 a 75.

Exercici 4 (1 punt)

Fer una funció anomenada Escrutini per saber si un cartró ha aconseguit Bingo. La funció rebrà un array que serà el cartró, la dimensió del cartró i un array (bombo) que contindrà informació dels noms que han sortit. En l'array bombó si el valor d'una determinada posició (index) és un 1, vindrà dir que la bola correspondent ha sortit i si el és un 0, vindrà dir que aquella bola no ha sortit.

La funció retornarà True en el cas que s'hagi aconseguit Bingo i False en cas contrari.

Exercici 5 (1 punt)

Exercici 5 (1 punt)

Fer un procediment anomenat ImprimirComentari per imprimir un comentari sobre el marcador. El procediment rebrà dos valors enters que seran el nombre de partides guanyades per cada jugador. En el cas que la diferència sigui:

- menor o igual que 1, escriure el missatge "Està molt igualat".
- entre 2 i 5, el missatge hauria de ser "La remuntada és possible".
- Maior o igual que 6, el missatge haurà de ser "Avui és el teu dia de sort".

NOTA: podeu utilitzar la funció abs() per obtenir el valor absolut d'un nombre.

Exercici 6 (2.5 punts)

Suposeu que teniu els següents procediments i funcions ja implementats:

```
void MenuPrincipal()
{
    cout << "--- Menu Principal ---" << endl;
    cout << "1.- Asignar cartrons" << endl;
    cout << "2.- Jugar" << endl;
    cout << "3.- Marcador" << endl;
    cout << "4.- Sortir" << endl;
}
```

Mostra per pantalla un menú amb les opcions del codi: 1. Asignar cartrons, 2. Jugar, etc.

void InicialitzarArray(int a[], int dim, int v)
Inicialitza el vector a de dimensió dim al valor v

int GenerarNombre(int min, int max, int a[], int dim)
Retorna un enter aleatori entre min i max, controlant que no estigui a l'array a de dimensió dim

Fer un programa complet (declaracions globals i funció main()) que segueixi els següents passos:

- Declarar (on correspongui) les constants necessàries per tal que el programa sigui fàcilment modificable.
- Declarar (on correspongui) els arrays necessaris per representar els cartrons de dos jugadors anomenats cartróPlayer1 i cartróPlayer2. Declarar també un array, anomenat bombo, per controlar quins nombres han sortit i quins no.
- Utilitzar el procediment MenuPrincipal() i llegir l'opció escollida.
- Si l'opció és 1, generar dos cartrons (un per a cada jugador) seguint els passos següents:
 - Inicialitzar els cartrons dels jugadors tot a 0 amb el procediment InicialitzarArray.
 - Utilitzar el procediment CrearCartro per crear els cartrons del jugador 1 i del jugador 2.
- Comprovar si els dos cartrons són diferents utilitzant la funció ArraysIguals. En cas que siguin iguals tornar a generar un dels dos.
- Si l'opció és 2, implementa el joc seguint els passos següents:
 - Si l'opció és 2, comprovar que s'han generat els cartrons (s'ha entrat a l'opció 1). En cas contrari, donar un missatge d'error i tornar al menú principal sense fer res més.
 - Inicialitzar l'array bombo a 0 amb el procediment InicialitzarArray.
 - Generar un nombre per simular la bola amb la funció GenerarNombre.
 - Imprimir el nombre que ha sortit i utilitzar la funció ImprimirBolesSortides per imprimir els noms de les boles que han sortit al llarg de la partida.
 - Utilitzar la funció Escrutini per comprovar si algun dels dos jugadors ha aconseguit fer Bingo.
 - En cas que no hi hagi cap guanyador tornar al punt 5.3
 - Si l'opció és 3, imprimir el marcador de les partides que s'han jugat i mostrar el missatge generat pel procediment ImprimirComentari.
 - Si l'opció és 4, imprimir el marcador final i mostrar el missatge: "Sortint del programa...".
 - Qualsevol altra opció, escriure el missatge: "Opció no permesa".
 - Repetir els passos 3 a 9, fins que l'opció del menú escollida sigui la 4.

Exercici 7 (0,5 punts)
Declarar un nou tipus de dades, `TCartró`, com un registre amb els camps: `Ident_Cartró` (valor enter), `Numeros` (array de 25 enters) i `Nom_Jugador` (cadena de 50 caràcters).

Fer un procediment anomenat `Imprimir` que rebi com a paràmetre un registre del tipus `TCartró` i imprimeix l'identificador del cartró i el nom del jugador separant-los amb el caràcter ‘`,`’.

Exercici 9 (1 punt)
Escriure la funció `Elevar` que retorni el resultat d'elevat un nombre al quadrat. La funció tindrà paràmetres de tipus `int`. El primer serà la base i el segon serà el resultat de l'operació en el cas que es pugui realizar. La funció retornarà un 1 si s'ha pogut realizar l'operació i un 0 en cas contrari.

NOTA: El valor de la base ha d'estar inclos en l'interval [-46.340, 46.340] per evitar overflow.

Exercici 8 (1 punt)

Fer una funció anomenada `Es_MatPositiva` que rebi com a paràmetres: una matriu d'enters de 4 columnes, el número de files de la matriu, i el número de columnes de la matriu, i retorna 1 si tots els valors de la matriu són positius o zero, i un 0 en cas contrari.

Examen Parcial (23 de Novembre de 2020)

Nom estudiant: Carolina Rose Kotter Grup: 45
NIU: 4055294

Fer la funció anomenada `ArraysIguals` per saber si dos arrays tenen tots els valors idèntics. La funció rebrà dos arrays i la seva dimensió. Retornarà `True` en el cas que siguin iguals i `False` en cas contrari.

NOTA: Podeu suposar que els dos arrays tenen la mateixa dimensió.

Important: Recordeu que cal donar les millors solucions possibles en cada exercici. A més de funcionar correctament, el codi ha d'estar ben programat (codi clar, amb les instruccions més adients, sense operacions ni variables innecessàries, etc.) Les funcions i procediments de les preguntes de la 1 a la 6 formen part d'un únic programa i, per tant, ha d'haver coherentia entre les seves definicions, i la forma en què s'utilitzen en altres preguntes. El context del programa a desenvoluppar, el tenuu a continuació:

El Bingo de 75 boles és un joc d'atzar on hi ha 75 boles numerades de l'1 al 75 dins del bombo. Els cartrons tenen un total de 25 nombres. En aquest tipus de bingo, es van treient boles del bombo i es guanya un premi quan han sortit tots els nombres del nostre cartró. Això es coneix com a bingo o full House.

Volem fer un programa que permeti jugar al Bingo a dos jugadors. Els jugadors podran jugar tantes partides com vulguin i caldrà portar el compte de quantes partides ha guanyat cadascú.

Exercici 1 (1 punt)

Fer un procediment anomenat `CrearCartro` per a generar un cartró. S'han de generar 25 nombres del 1 al 75. S'ha d'assegurar que hi hagin:

- 5 nombres del 11 al 15,
- 5 nombres del 46 al 60 i
- 5 nombres del 61 al 75.
- 5 nombres del 16 al 30,
- 5 nombres del 31 al 45,

El procediment rebrà un array (que serà el cartró) i la dimensió de l'array. Retornarà l'array ple amb els números ordenats de menor a major.

UTILITZEU (no implementar) els següents procediments o funcions:

- `int GenerarNombre (int min, int max, int a[], int dim)`
Genera un nombre aleatori entre min i max, i revisa que no estigui a l'array a de dimensió dim.
- `void Ordena (int a[], int dim)`
Ordena l'array a de dimensió dim de menor a major.

$$20 \times 10 = 200$$

$$200 + 50 = \underline{\underline{250}}$$

$$4 \times 3 = 10$$

$$10 \times 2 = 20$$

Ressolt: 20

Exercici 3 (1 punt)

Fer un procediment anomenat `ImprimirBolesSortides` per imprimir els nombres de les boles que ja han sortit. El procediment rebrà un array i la seva dimensió. L'array contindrà informació sobre si un nombre ha sortit o no. Si el valor d'una determinada posició (index) és un 1, voldrà dir que la bola corresponent ha sortit i si el és un 0, voldrà dir que aquella bola no ha sortit.

NOTA: Recordeu que els números de les boles van de 1 a 75.

Exercici 4 (1 punt)
Fer una funció anomenada Escrutini per saber si un cartó ha aconseguit Bingo. La funció rebrà un array que serà el cartó, la dimensió del cartó i un array (bombo) que contindrà informació dels nombres que han sortit. En l'array bombo si el valor d'una determinada posició (index) és un 1, vindrà dir que la bola corresponent ha sortit i si el és un 0, vindrà dir que aquella bola no ha sortit.

La funció retornarà true en el cas que s'hagi aconseguit Bingo i false en cas contrari.

Exercici 6 (2.5 punts)
Suposeu que teniu els següents procediments i funcions ja implementats:

```
void MenuPrincipal()
{
    cout << " --- Menú Principal --- " << endl;
    cout << "1. - Asignar cartones " << endl;
    cout << "2. - Jugar" << endl;
    cout << "3. - Marcador" << endl;
    cout << "4. - Sortir" << endl;
}

Mostra per pantalla un menú amb les opcions del codi: 1. Asignar cartons, 2. Jugar, etc.
```

```
void InicialitzarArray(int min, int max, int a[], int dim)
Inicialitza el vector a de dimensió dim al valor v
```

```
int GenerarNombre(int min, int max, int v)
```

Retorna un enter aleatori entre min i max, controlant que no estigui a l'array a de dimensió dim

Fer un programa complet (declaracions globals i funció main()) que segueixi els següents passos:

1. Declarar (on correspongui) les constants necessàries per tal que el programa sigui fàcilment modificable.
2. Declarar (on correspongui) els arrays necessaris per representar els cartons de dos jugadors anomenats cartoPlayer1 i cartoPlayer2. Declarar també un array, anomenat bombo, per controlar quins noms han sortit quins no.
3. Utilitzar el procediment MenuPrincipal() i llegir l'opció escollida.
4. Si l'opció és 1, generar dos cartons (un per a cada jugador), seguint els passos següents:
 - 4.1. Inicialitzar els cartons dels jugadors tot a 0 amb el procediment InicialitzarArray.
 - 4.2. Utilitzar el procediment CrearCartro per crear els cartons del jugador 1 i del jugador 2.
- 4.3. Comprovar si els dos cartons són diferents utilitzant la funció ArraysIguals. En cas que siguin iguals tornar a generar un dels dos.
5. Si l'opció és 2, implementa el joc seguint els passos següents:
 - 5.1. S'haurà de comprovar que s'han generat els cartons (s'ha entrat a l'opció 1). En cas contrari, donar un missatge d'error i tornar al menú principal sense fer res més.
 - 5.2. Inicialitzar l'array bombo a 0 amb el procediment InicialitzarArray.
 - 5.3. Generar un nombre per simular la bola amb la funció GenerarNombre.
 - 5.4. Imprimir el nombre que ha sortit i utilitzar la funció ImprimirBolesSortides per imprimir els noms de les boles que han sortit al llarg de la partida.
 - 5.5. Utilitzar la funció Escrutini per comprovar si algun dels dos jugadors ha aconseguit fer Bingo.
 - 5.6. En cas que no hi hagi cap guanyador tornar al punt 5.3
 6. Si l'opció és 3: imprimir el marcador de les partides que s'han jugat i mostrar el missatge generat pel procediment ImprimirComentari.
 7. Si l'opció és 4, imprimir el marcador final i mostrar el missatge: "Sortint del programa..."
 8. Qualsevol altra opció, escriure el missatge: "Opció no permesa".
 9. Repetir els passos 3 a 9, fins que l'opció del menú escollida sigui la 4.

Declarar un nou tipus de dades, TCartro, com un registre amb els camps: Ident_Cartro (valor enter), Numeros (array de 25 enters) i Nom_Jugador (cadena de 50 caràcters).
Fer un procediment anomenat Imprimir que rebi com a paràmetre un registre del tipus TCartro i imprimeix l'identificador del cartó i el nom del jugador separant -los amb el caràcter '-'.

Exercici 7 (1 punt)
Escriviu la funció Elevar que retorna el resultat d'elevar un nombre a quadrat. La funció tindrà dos paràmetres de tipus int. El primer serà la base i el segon serà el resultat de l'operació en el cas que es pugui realizar. La funció retornarà un 1 si s'ha pogut realitzar l'operació i un 0 en cas contrari.
NOTA: El valor de la base ha d'estar inclos en l'interval [-46.340,46.340] per evitar overflow.

Exercici 8 (1 punt)

Fer una funció anomenada Es_MatPositiva que rebi com a paràmetres: una matriu d'enters de 4 columnes, el número de files de la matriu, i el número de columnes de la matriu, i retorni 1 si tots els valors de la matriu són positius o zero, i un 0 en cas contrari.

Examen Parcial (23 de Novembre de 2020)**Nom estudiant:** Claudia Miquel**Grup:** NIU: 4091694

Exercici 1 (1 punt)
 Fer la funció anomenada `ArraysIguals` per saber si dos arrays tenen tots els valors idèntics. La funció rebrà dos arrays i la seva dimensió. Retornarà `True` en el cas que siguin iguals i `False` en cas contrari.

NOTA: Podeu suposar que els dos arrays tenen la mateixa dimensió.

Important: Recordeu que cal donar les millors solucions possibles en cada exercici. A més de funcionar correctament, el codi ha d'estar ben programat (codi clar, amb les instruccions més adients, sense operacions ni variables innecessàries, etc.) Les funcions i procediments de les preguntes de la 1 a la 6 formen part d'un únic programa i, per tant, ha d'haver coherència entre les seves definicions, i la forma en què s'utilitzen en altres preguntes. El context del programa a desenvolupar, el teniu a continuació:

El Bingo de 75 boles és un joc d'atzar on hi ha 75 boles numerades de l'1 al 75 dins del bumbo. Els cartells tenen un total de 25 números. En aquest tipus de bingo, es van treient boles del bumbo i es guanya un premi quan han sortit tots els números del nostre cartell. Això es coneix com a bingo o full House.

Volem fer un programa que permeti jugar al Bingo a dos jugadors. Els jugadors podran jugar tantes partides com vulguin i caldrà portar el compte de quantes partides ha guanyat cadascú.

Exercici 1 (1 punt)

Fer un procediment anomenat `CrearCartell` per a generar un cartell. S'han de generar 25 nombres del 1 al 75. S'ha d'assegurar que hi hagin:

- 5 nombres del 11 al 15.
- 5 nombres del 46 al 60 i
- 5 nombres del 61 al 75.
- 5 nombres del 16 al 30.
- 5 nombres del 31 al 45,

El procediment rebrà un array (que serà el cartell) i la dimensió de l'array. Retornarà l'array ple amb els números ordenats de menor a major.

UTILITZEU (no implementar) els següents procediments o funcions:

- `int GenerarNombre (int min, int max, int a[], int dim)`
Genera un nombre aleatori entre min i max, i revisa que no estigui a l'array a de dimensió dim.
- `void Ordena (int a[], int dim)`
Ordena l'array a de dimensió dim de menor a major.

$$20 \times 10 = 200$$

$$200 + 50 = 250$$

$$4 \times 3 = 12$$

$$\text{Resultat: } 38$$

Exercici 3 (1 punt)

Fer un procediment anomenat `ImprimirBolesSortides` per imprimir els nombres de les boles que ja han sortit. El procediment rebrà un array i la seva dimensió. L'array contingrà informació sobre si un nombre ha sortit o no. Si el valor d'una determinada posició (index) es un 1, voldrà dir que la bola corresponent ha sortit i si el és un 0, voldrà dir que aquella bola no ha sortit.

NOTA: Recordeu que els números de les boles van de 1 a 75.

Exercici 4 (1 punt)
Fer una funció anomenada Escrutini per saber si un cartró ha aconseguit Bingo. La funció rebrà un array que serà el cartró, la dimensió del cartró i un array (bombo) que contindrà informació dels nombres que han sortit. En l'array bombo si el valor d'una determinada posició (index) és un 1, voldrà dir que la bola corresponent ha sortit i si el és un 0, voldrà dir que aquella bola no ha sortit.

La funció retornarà true en el cas que s'hagi aconseguit Bingo i False en cas contrari.

```
}
```

Exercici 5 (2,5 punts)
Suposeu que teniu els següents procediments i funcions ja implementats:

```
void MenuPrincipal()
{
    cout << "---- Menu Principal ----" << endl;
    cout << "1.- Asignar cartones" << endl;
    cout << "2.- Jugar" << endl;
    cout << "3.- Marcador" << endl;
    cout << "4.- Sortir" << endl;
}
```

Mostra per pantalla un menú amb les opcions del codi: 1. Asignar cartons, 2. Jugar, etc.

```
void InicialitzarArray(int a[], int dim, int v)
```

Inicialitza el vector a de dimensió dim al valor v

```
int GenerarNombre (int min, int max, int a[], int dim)
```

Retorna un enter aleatori entre min i max, controlant que no estigui a l'array a de dimensió dim

Fer un programa complet (declaracions globals i funció main()) que segueixi els següents passos:

1. Declarar (on correspongui) les constants necessàries per tal que el programa sigui fàcilment modificable.
2. Declarar (on correspongui) els arrays necessaris per representar els cartrons de dos jugadors anomenats cartróPlayer1 i cartróPlayer2. Declarar també un array, anomenat bombo, per controlar quins noms han sortit i quins no.
3. Utilitzar el procediment MenuPrincipal () i llegir l'opció escollida.
4. Si l'opció és 1, generar dos cartrons (un per a cada jugador) seguint els passos següents:
 - 4.1. Inicialitzar els cartrons dels jugadors tot a 0 amb el procediment InicialitzarArray.
 - 4.2. Utilitzar el procediment CrearCartró per crear els cartrons del jugador 1 i del jugador 2.
 - 4.3. Comprovar si els dos cartrons són diferents utilitzant la funció ArraysIguals. En cas que siguin iguals tornar a generar un dels dos.
5. Si l'opció és 2, implementa el joc seguint els passos següents:
 - 5.1. Shaurà de comprovar que s'han generat els cartrons (s'ha entrat a l'opció 1). En cas contrari, donar un missatge d'error i tornar al menú principal sense fer res més.
 - 5.2. Inicialitzar l'array bombo a 0 amb el procediment InicialitzarArray.
 - 5.3. Generar un nombre per simular la bola amb la funció GenerarNombre.
 - 5.4. Imprimir el nombre que ha sortit i utilitzar la funció ImprimirBolesSortides per imprimir els noms de les boles que han sortit al llarg de la partida.
 - 5.5. Utilitzar la funció Escrutini per comprovar si algun dels dos jugadors ha aconseguit fer Bingo.
 - 5.6. En cas que no hi hagi cap guanyador tornar al punt 5.3
 6. Si l'opció és 3: imprimir el marcador de les partides que s'han jugat i mostrar el missatge generat pel procediment ImprimirComentari.
 7. Si l'opció és 4, imprimir el marcador final i mostrar el missatge: "Sortint del programa..."
 8. Qualsevol altra opció, escriure el missatge: "Opció no permesa".
 9. Repetir els passos 3 a 9, fins que l'opció del menú escollida sigui la 4.

Declarar un nou tipus de dades, TCartró, com un registre amb els camps: Ident_Cartró (valor enter), Numeros (array de 25 enters) i Nom_Jugador (cadena de 50 caràcters).
Fer un procediment anomenat Imprimir que rebi com a paràmetre un registre del tipus TCartró i imprimeix l'identificador del cartró i el nom del jugador separant-los amb el caràcter ' '.

Escriure la funció Elevar que retorna el resultat de elevar un nombre al quadrat. La funció tindrà dos paràmetres de tipus int. El primer serà la base i el segon serà el resultat de l'operació en el cas que es pugui realitzar. La funció retornarà un 1 si s'ha pogut realitzar l'operació i un 0 en cas contrari.

NOTA: El valor de la base ha d'estar inclos en l'intervall [-46.340, 46.340] per evitar overflow.

Exercici 8 (1 punt)

Fer una funció anomenada Es_MatPositiva que rebi com a paràmetres: una matriu d'enters de 4 columnes, el número de files de la matriu i el número de columnes de la matriu, i retorna 1 si tots els valors de la matriu són positius o zero, i un 0 en cas contrari.

Exercicis d'Informàtica (103806)

Curs 2020-2021
Examen Parcial (23 de Novembre de 2020)

Nom estudiant: CASSIE SANDOVAL

NIU: 8565110

Important: Recordeu que cal donar les millors solucions possibles en cada exercici. A més de funcionar correctament, el codi ha d'estar ben programat (codi clar, amb les instruccions més adients, sense operacions ni variables innecessàries, etc.)

Les funcions i procediments de les preguntes de la 1 a la 6 formen part d'un únic programa i, per tant, ha d'haver coherentia entre les seves definicions, i la forma en què s'utilitzen en altres preguntes. El context del programa a desenvolupar, el teniu a continuació:

El Bingo de 75 boles és un joc d'atzar on hi ha 75 boles numerades de l'1 al 75 dins del bombo. Els cartrons tenen un total de 25 nombres. En aquest tipus de bingo, es van treient boles del bombo i es guanya un premi quan han sortit tots els nombres del nostre cartró. Això es coneix com a bingo o full House.

Volem fer un programa que permeti jugar al Bingo a dos jugadors. Els jugadors podran jugar tantes partides com vulguin i caldrà portar el compte de quantes partides ha guanyat cadascú.

Exercici 1 (1 punt)

Fer un procediment anomenat CrearCartro per a generar un cartró. S'han de generar 25 nombres del 1 al 75. S'ha d'assegurar que hi hagin:

- 5 nombres del 11 al 15,
- 5 nombres del 16 al 30,
- 5 nombres del 31 al 45,
- 5 nombres del 46 al 60 i
- 5 nombres del 61 al 75.

El procediment rebrà un array (que serà el cartró) i la dimensió de l'array. Retornarà l'array ple amb els números ordenats de menor a major.

UTILITZEU (no implementar) els següents procediments o funcions:

- int GenerarNombre (int min, int max, int a[], int dim)
Genera un nombre aleatori entre min i max, i revisa que no estigui a l'array a de dimensió dim.
- void Ordena (int a[], int dim)
Ordena l'array a de dimensió dim de menor a major.

$$20 \times 20 = 400$$

$$7 \div 7 = \boxed{1}$$

Exercici 2 (1 punt)
Fer la funció anomenada ArraysIguals per saber si dos arrays tenen tots els valors idèntics. La funció rebrà dos arrays i la seva dimensió. Retornarà true en el cas que siguin iguals i false en cas contrari.

NOTA: Podeu suposar que els dos arrays tenen la mateixa dimensió.

$$300 + 300 = 600$$
$$150 \times 2 = 300$$

$$\text{RESUM} = 100$$

Exercici 3 (1 punt)

Fer un procediment anomenat ImprimirBolesSortides per imprimir els nombres de les boles que ja han sortit. El procediment rebrà un array i la seva dimensió. L'array contingdrà informació sobre si un nombre ha sortit o no. Si el valor d'una determinada posició (index) és un 1, voldrà dir que la bola corresponent ha sortit i si el és un 0, voldrà dir que aquella bola no ha sortit.

NOTA. Recordeu que els números de les boles van de 1 a 75.

Exercici 4 (1 punt)

Fer una funció anomenada `Escrutini` per saber si un cartró ha aconseguit Bingo. La funció rebrà un array que sera el cartró, la dimensió del cartró i un array (bombo) que contindrà informació dels nombres que han sortit. En l'array bombó si el valor d'una determinada posició (`index`) és un 1, vindrà dir que la bola correspondent ha sortit i si el és un 0, vindrà dir que aquella bola no ha sortit.

La funció retornarà `true` en el cas que s'hagi aconseguit Bingo i `false` en cas contrari.

```
}
```

Mostra per pantalla un menú amb les opcions del codi: 1. Assignar cartrons, 2. Jugar, etc.

void InicialitzarArray(int a[], int dim, int v)

Retorna un enter aleatori entre min i max, controlant que no estigui a l'array a de dimensió dim

```
int GenerarNombre (int min, int max, int a[], int dim)
```

Inicialitza el vector a de dimensió dim al valor v

Exercici 6 (2.5 punts)

Suposeu que teniu els següents procediments i funcions ja implementats:

```
void MenuPrincipal()
{
    cout << " --- Menu Principal --- " << endl;
    cout << "1.- Asignar cartrons " << endl;
    cout << "2.- Jugar" << endl;
    cout << "3.- Marcador" << endl;
    cout << "4.- Sortir" << endl;
}
```

Mostra per pantalla un menú amb les opcions del codi: 1. Assignar cartrons, 2. Jugar, etc.

void InicialitzarArray(int a[], int dim, int v)

Retorna un enter aleatori entre min i max, controlant que no estigui a l'array a de dimensió dim

```
int GenerarNombre (int min, int max, int a[], int dim)
```

Inicialitza el vector a de dimensió dim al valor v

Fer un programa complet (declaracions globals i funció main) que segueixi els següents passos:

1. Declarar (on corresponguï) les constants necessàries per tal que el programa sigui fàcilment modificable.
2. Declarar (on corresponguï) els arrays necessaris per representar els cartrons de dos jugadors anomenats cartroplayer1 i cartroplayer2. Declarar també un array, anomenat bombo, per controlar quins noms han sortit i quins no.
3. Utilitzar el procediment `MenuPrincipal()` i llegir l'opció escollida.
4. Si l'opció és 1, generar dos cartrons (un per a cada jugador) seguint els passos següents:
 - 4.1. Inicialitzar els cartrons dels jugadors tot a 0 amb el procediment `InicialitzarArray`.
 - 4.2. Utilitzar el procediment `CrearCartró` per crear els cartrons del jugador 1 i del jugador 2.
 - 4.3. Comprovar si els dos cartrons són diferents utilitzant la funció `ArraysIguals`. En cas que siguin iguals tornar a generar un dels dos.
5. Si l'opció és 2, implementa el joc següent els passos següents:
 - 5.1. S'haurà de comprovar que s'han generat els cartrons (s'ha entrat a l'opció 1). En cas contrari, donar un missatge d'error i tornar al menú principal sense fer res més.
 - 5.2. Inicialitzar l'array bombo a 0 amb el procediment `InicialitzarArray`.
 - 5.3. Generar un nombre per simular la bola amb la funció `GenerarNombre`.
 - 5.4. Imprimir el nombre que ha sortit i utilitzar la funció `ImprimirBolesSortides` per imprimir els noms de les boles que han sortit al llarg de la partida.
 - 5.5. Utilitzar la funció `Escrutini` per comprovar si algun dels dos jugadors ha aconseguit fer Bingo.
 - 5.6. En cas que no hi hagi cap guanyador tornar al punt 5.3
6. Si l'opció és 3: imprimir el marcador de les partides que s'han jugat i mostrar el missatge generat pel procediment `ImprimirComentari`.
7. Si l'opció és 4, imprimir el marcador final i mostrar el missatge: "Sortint del programa..."
8. Qualsevol altra opció, escriure el missatge: "Opció no permesa".
9. Repetir els passos 3 a 9, fins que l'opció del menú escollida sigui la 4.

Exercici 7 (0,5 punts)

Declarar un nou tipus de dades, `TCartro`, com un registre amb els camps: `Ident_Cartro` (valor enter), `Numeros` (array de 25 enters) i `Nom_Jugador` (cadena de 50 caràcters).
Fer un procediment anomenat `Imprimir` que rebi com a paràmetre un registre del tipus `TCartro` i imprimeixi l'identificador del cartró i el nom del jugador separant-los amb el caràcter `'.'`.

Exercici 9 (1 punt)

Escrivre la funció `Elevar` que retorni el resultat de elevar un nombre al quadrat. La funció tindrà 2 paràmetres de tipus int. El primer serà la base i el segon serà el resultat de l'operació en el cas que es pugui realizar. La funció retornarà un 1 si s'ha pogut realitzar l'operació i un 0 en cas contrari.

NOTA: El valor de la base ha d'estar inclos en l'intervall [-46.340,46.340] per evitar overflow.

Exercici 8 (1 punt)

Fer una funció anomenada `Es_MatPositiva` que rebi com a paràmetres: una matriu d'enters de 4 columnes, el número de files de la matriu, i el número de columnes de la matriu, i retorna 1 si tots els valors de la matriu són positius o zero, i un 0 en cas contrari.

Examen Parcial (23 de Novembre de 2020)

Curs 2020-2021
Grup:
NIU: 242652

Nom estudiant: SOE CARTEZ

Exercici 2 (1 punt)
Fer la funció anomenada `ArraysIguals` per saber si dos arrays tenen tots els valors idèntics.
La funció rebrà dos arrays i la seva dimensió. Retornarà `True` en el cas que siguin iguals i `False` en cas contrari.

NOTA: Podeu suposar que els dos arrays tenen la mateixa dimensió.

Important: Recordeu que cal donar les millors solucions possibles en cada exercici. A més de funcionar correctament, el codi ha d'estar ben programat (codi clar, amb les instruccions més adients, sense operacions ni variables innecessàries, etc.)
Les funcions i procediments de les preguntes de la 1 a la 6 formen part d'un únic programa i, per tant, ha d'haver coherència entre les seves definicions, i la forma en què s'utilitzen en altres preguntes. El context del programa a desenvolupar, el teniu a continuació:

El Bingo de 75 boles és un joc d'atzar on hi ha 75 boles numerades de l'1 al 75 dins del bombo. Els cartells tenen un total de 25 nombres. En aquest tipus de bingo, es van treient boles del bombo i es guanya un premi quan han sortit tots els nombres del nostre cartell. Això es coneix com a bingo o full House.

Volem fer un programa que permeti jugar al Bingo a dos jugadors. Els jugadors podran jugar tantes partides com vulguin i caldrà portar el compte de quantes partides ha guanyat cadaescú.

$$6 \times 6 = 36 \\ \text{REULTAT} = 12$$

Exercici 1 (1 punt)
Fer un procediment anomenat `CrearCartell` per a generar un cartell. S'han de generar 25 nombres del 1 al 75. S'ha d'assegurar que hi hagin:

- 5 nombres del 11 al 15,
- 5 nombres del 16 al 30,
- 5 nombres del 31 al 45,
- 5 nombres del 46 al 60 i
- 5 nombres del 61 al 75.

El procediment rebrà un array (que serà el cartell) i la dimensió de l'array. Retornarà l'array ple amb els números ordenats de menor a major.

UTILITZEU (no implementar) els següents procediments o funcions:

- `int GenerarNombre(int min, int max, int a[], int dim)`
Genera un nombre aleatori entre `min` i `max`, i revisa que no estigui a l'array `a` de dimensió `dim`.
- `void Ordena(int a[], int dim)`
Ordena l'array `a` de dimensió `dim` de menor a major.

$$25 + 13 = 38 \\ 10 + 2 = \underline{\underline{12}}$$

Exercici 4 (1 punt)

Fer una funció anomenada `Escrutini` per saber si un cartró ha aconseguit Bingo. La funció rebrà un array que serà el cartró, la dimensió del cartró i un array (bombo) que contingrà informació dels noms de les boles que han sortit. En l'array bombo si el valor d'una determinada posició (`index`) és un 1, vindrà dir que la bola corresponent ha sortit i si el és un 0, vindrà dir que aquella bola no ha sortit.

La funció retornarà `True` en el cas que s'hagi aconseguit Bingo i `False` en cas contrari.

`bool Escrutini(int a[], int dim, int b[]);`

Exercici 6 (2,5 punts)

Suposeu que teniu els següents procediments i funcions ja implementats:

```
void Menuprincipal()
{
    cout << " --- Menu Principal --- " << endl;
    cout << "1.- Asignar cartrons " << endl;
    cout << "2.- Jugar " << endl;
    cout << "3.- Marcador" << endl;
    cout << "4.- Sortir" << endl;
}
```

Mostra per pantalla un menú amb les opcions del codi: 1. Asignar cartrons, 2. Jugar, etc.

voíd InicialitzarArray(int a[], int dim, int v)

Inicialitza el vector a de dimensió dim al valor v

int GenerarNombre (int min, int max, int a[], int dim)

Retorna un enter aleatori entre min i max, controlant que no estigui a l'array a de dimensió dim

Fer un programa complet (declaracions globals i funció main()) que segueixi els següents passos:

1. Declarar (on correspongui) les constants necessàries per tal que el programa sigui fàcilment modificable.
2. Declarar (on correspongui) els arrays necessaris per representar els cartrons de dos jugadors anomenats `cartroplayer1` i `cartroplayer2`. Declarar també un array, anomenat `bombo`, per controlar quins nombres han sortit i quins no.
3. Utilitzar el procediment `Menuprincipal()` i llegir l'opció escollida.
4. Si l'opció és 1, generar dos cartrons (un per a cada jugador) seguint els passos següents:
 - 4.1. Inicialitzar els cartrons dels jugadors tot a 0 amb el procediment `InicialitzarArray`.
 - 4.2. Utilitzar el procediment `CrearCartro` per crear els cartrons del jugador 1 i del jugador 2.
 - 4.3. Comprovar si els dos cartrons són diferents utilitzant la funció `ArraysIguals`. En cas que siguin iguals tornar a generar un dels dos.
5. Si l'opció és 2, implementa el joc seguint els passos següents:
 - 5.1. S'haurà de comprovar que s'han generat els cartrons (s'ha entrat a l'opció 1). En cas contrari, donar un missatge d'error i tornar al menú principal sense fer res més.
 - 5.2. Inicialitzar l'array `bombo` a 0 amb el procediment `InicialitzarArray`.
 - 5.3. Generar un nombre per simular la bola amb la funció `GenerarNombre`.
 - 5.4. Imprimir el nombre que ha sortit i utilitzar la funció `ImprimirBolesSortides` per imprimir els noms de les boles que han sortit al llarg de la partida.
 - 5.5. Utilitzar la funció `Escrutini` per comprovar si algun dels dos jugadors ha aconseguit fer Bingo.
 - 5.6. En cas que no hi hagi cap guanyador tornar al punt 5.3
6. Si l'opció és 3: imprimir el marcador de les partides que s'han jugat i mostrar el missatge generat pel procediment `ImprimirComentari`.
7. Si l'opció és 4, imprimir el marcador final i mostrar el missatge: "Sortint del programa...".
8. Qualsevol altra opció, escriure el missatge: "Opció no permesa".
9. Repetir els passos 3 a 9, fins que l'opció del menú escollida sigui la 4.

Exercici 7 (0,5 punts)

Declarar un nou tipus de dades, `TCartro`, com un registre amb els camps: `Ident_Cartro` (valor enter), `Numeros` (array de 25 enters) i `nom_Jugador` (cadena de 50 caràcters).
Fer un procediment anomenat `Imprimir` que rebi com a paràmetre un registre del tipus `TCartro` i imprimeixi l'identificador del cartró i el nom del jugador separant-los amb el caràcter `'`.

Exercici 9 (1 punt)

Escriure la funció `Elevar` que retorni el resultat d'elevat un nombre al quadrat. La funció tindrà 2 paràmetres de tipus int. El primer serà la base i el segon serà el resultat de l'operació en el cas que es pugui realizar. La funció retornarà un 1 si s'ha pogut realizar l'operació i un 0 en cas contrari.

NOTA: El valor de la base ha d'estar inclos en l'intervall [-46,340,46,340] per evitar overflow.

Exercici 8 (1 punt)

Fer una funció anomenada `E8_MatPositiva` que rebi com a paràmetres: una matriu d'enters de 4 columnes, el número de files de la matriu, i el número de columnes de la matriu, i retorna 1 si tots els valors de la matriu són positius o zero, i un 0 en cas contrari.

Onaments d'Informàtica (103806)

Curs 2020-2021

Examen Parcial (23 de Novembre de 2020)

Grup: NIU: 6528150

Nom estudiant: BOUVE THOMAS JORDI

Important: Recordeu que cal donar les millors solucions possibles en cada exercici. A més de funcionar correctament, el codi ha d'estar ben programat (codi clar, amb les instruccions més adients, sense operacions ni variables innecessàries, etc.)

Les funcions i procediments de les preguntes de la 1 a la 6 formen part d'un únic programa i, per tant, ha d'haver coherència entre les seves definicions, i la forma en què s'utilitzen en altres preguntes. El context del programa a desenvolupar, el teniu a continuació:

El Bingo de 75 boles és un joc d'atzar on hi ha 75 boles numerades de 1 al 75 dins del bombo. Els cartells tenen un total de 25 nombres. En aquests tipus de bingo, es van treient boles del bombo i es guanya un premi quan han sortit tots els nombres del nostre cartell. Això es coneix com a bingo o full House.

Volem fer un programa que permeti jugar al Bingo a dos jugadors. Els jugadors podran jugar tantes partides com vulguin i caldrà portar el compte de quantes partides ha guanyat cadascú.

Exercici 1 (1 punt)

Fer un procediment anomenat CrearCartell per a generar un cartell. S'han de generar 25 nombres del 1 al 75. S'ha d'assegurar que hi hagin:

- 5 nombres del 1 al 15,
- 5 nombres del 16 al 30,
- 5 nombres del 31 al 45,
- 5 nombres del 46 al 60 i
- 5 nombres del 61 al 75.

El procediment rebrà un array (que serà el cartell) i la dimensió de l'array. Retornarà l'array ple amb els números ordenats de menor a major.

UTILITZEU (no implementar) els següents procediments o funcions:

- int GeneraNombre (int min, int max, int a[], int dim).
Genera un nombre aleatori entre min i max, i revisa que no estigui a l'array a de dimensió dim.
- void Ordena (int a[], int dim).
Ordena l'array a de dimensió dim de menor a major.

$$17 + 17 = \underline{\underline{34}}$$

Exercici 2 (1 punt)
Fer la funció anomenada ArraysIguals per saber si dos arrays tenen tots els valors idèntics. La funció rebrà dos arrays i la seva dimensió. Retornarà True en el cas que siguin iguals i False en cas contrari.

NOTA: Podeu suposar que els dos arrays tenen la mateixa dimensió.

$$45 + 45 = 90$$

$$113 - 50 = 2$$

Exercici 3 (1 punt)
Fer un procediment anomenat ImprimirBolesSortides per imprimir els nombres de les boles que ja han sortit. El procediment rebrà un array i la seva dimensió. L'array contingrà informació sobre si un nombre ha sortit o no. Si el valor d'una determinada posició (index) és un 1, voltrà dir que la bola corresponent ha sortit i si el és un 0, voltrà dir que aquella bola no ha sortit.

NOTA: Recordeu que els números de les boles van de 1 a 75.

Exercici 4 (1 punt)

Fer una funció anomenada Escrutini per saber si un cartró ha aconseguit Bingo. La funció rebrà un array que serà el cartró, la dimensió del cartró i un array (bombo) que contindrà informació dels noms de les boles que han sortit. En l'array bombo si el valor d'una determinada posició (index) és un 1, vindrà dir que la bola corresponent ha sortit i si el és un 0, vindrà dir que aquella bola no ha sortit.

La funció retornarà True en el cas que s'hagi aconseguit Bingo i False en cas contrari.

Exercici 6 (2,5 punts)

Fer un programa complet (declaracions globals i funció main()) que segueixi els següents passos:

1. Declarar (on corresponguï) les constants necessàries per tal que el programa sigui fàcilment modificable.
2. Declarar (on corresponguï) els arrays necessaris per representar els cartons de dos jugadors anomenats cartroPlayer1 i cartroPlayer2. Declarar també un array, anomenat bombo, per controlar quins noms de les boles han sortit i quins no.
3. Utilitzar el procediment MenuPrincipal() i llegir l'opció escollida.

4. Si l'opció és 1, generar dos cartons (un per a cada jugador) seguint els passos següents:
 - 4.1. Inicialitzar els cartons dels jugadors tot a 0 amb el procediment InicialitzarArray.
 - 4.2. Utilitzar el procediment CrearCartro per crear els cartons del jugador 1 i del jugador 2.
- 4.3. Comprovar si els dos cartons són diferents utilitzant la funció ArraysIguals. En cas que siguin iguals tornar a generar un dels dos.

5. Si l'opció és 2, implementa el joc següint els passos següents:
 - 5.1. S'haurà de comprovar que s'han generat els cartons (s'ha entrat a l'opció 1). En cas contrari, donar un missatge d'error i tornar al menú principal sense fer res més.
 - 5.2. Inicialitzar l'array bombo a 0 amb el procediment InicialitzarArray.
 - 5.3. Generar un nombre per simular la bola amb la funció GenerarNombre.

- 5.4. Imprimir el nombre que ha sortit i utilitzar la funció ImprimirBolesSortides per imprimir els noms de les boles que han sortit al llarg de la partida.
- 5.5. Utilitzar la funció Escrutini per comprovar si algun dels dos jugadors ha aconseguit fer Bingo.

- 5.6. En cas que no hi hagi cap guanyador tornar al punt 5.3
6. Si l'opció és 3: imprimir el marcador de les partides que s'han jugat i mostrar el missatge generat pel procediment ImprimirComentari.

7. Si l'opció és 4, imprimir el marcador final i mostrar el missatge: "Sortint del programa..."
8. Qualsevol altra opció, escriure el missatge: "Opció no permesa".

9. Repetir els passos 3 a 9, fins que l'opció del menú escollida sigui la 4.

Mostra per pantalla un menú amb les opcions del codi: 1. Assignar cartons, 2. Jugar, etc.

```
void InicialitzarArray(int a[], int dim, int v)
    Inicialitza el vector a de dimensió dim al valor v
```

```
int GenerarNombre(int min, int max, int a[], int dim)
    Retorna un enter aleatori entre min i max, controlant que no estigui a l'array a de dimensió dim
```

Exercici 5 (1 punt)

Fer un procediment anomenat ImprimirComentari per imprimir un comentari sobre el marcador. El procediment rebrà dos valors enters que seran el nombre de partides guanyades per cada jugador. En el cas que la diferència sigui:

- menor o igual que 1, escriure el missatge "Està molt igualat".
- entre 2 i 5, el missatge hauria de ser "Avui és el teu dia de sort".
- Major o igual que 6, el missatge haurà de ser "La remuntada és possible".

NOTA: podeu utilitzar la funció abs() per obtenir el valor absolut d'un nombre.

Exercici 7 (0,5 punts)

Declarar un nou tipus de dades, `TCartro`, com un registre amb els camps: `Ident_Cartro` (valor enter), `Numeros` (array de 25 enters) i `Nom_Jugador` (cadena de 50 caràcters).

Fer un procediment anomenat `Imprimir` que rebi com a paràmetre un registre del tipus `TCartro` i imprimeix l'identificador del cartró i el nom del jugador separant-los amb el caràcter `,`.

Exercici 9 (1 punt)

Escriure la funció `Elevar` que retorna el resultat d'elevat un nombre al quadrat. La funció tindrà 2 paràmetres del tipus `int`. El primer serà la base i el segon serà el resultat de l'operació en el cas que es pugui realizar. La funció retornarà un 1 si s'ha pogut realizar l'operació i un 0 en cas contrari.

NOTA: El valor de la base ha d'estar inclos en l'interval [-46,340;46,340] per evitar overflow.

Exercici 8 (1 punt)

Fer una funció anomenada `E.S_MatPositiva` que rebi com a paràmetres: una matriu d'enters de 4 columnes, el número de files de la matriu, i el número de columnes de la matriu, i retorna 1 si tots els valors de la matriu son positius o zero, i un 0 en cas contrari.

Onaments d'Informàtica (103806)

Curs 2020-2021

Examen Parcial (23 de Novembre de 2020) Grup:

Nom estudiant: Heather Goldstein NIU: 2987229

Important: Recordeu que cal donar les millors solucions possibles en cada exercici. A més de funcionar correctament, el codi ha d'estar ben programat (codi clar, amb les instruccions més adients, sense operacions ni variables innecessàries, etc.)

Les funcions i procediments de les preguntes de la 1 a la 6 formen part d'un únic programa i, per tant, ha d'haver coherentia entre les seves definicions, i la forma en què s'utilitzen en altres preguntes. El context del programa a desenvolupar, el teniu a continuació:

El Bingo de 75 boles és un joc d'atzar on hi ha 75 boles numerades de l'1 al 75 dins del bombo. Els cartons tenen un total de 25 nombres. En aquest tipus de bingo, es van treient boles del bombo i es guanya un premi quan han sortit tots els nombres del nostre cartó. Això es coneix com a bingo o full House.

Volem fer un programa que permeti jugar al Bingo a dos jugadors. Els jugadors podran jugar tantes partides com vulguin i caindrà portar el compte de quantes partides ha guanyat cadaescú.

Exercici 1 (1 punt)
Fer un procediment anomenat CrearCartró per a generar un cartó. S'han de generar 25 nombres del 1 al 75. S'ha d'assegurar que hi hagin:

- 5 nombres del 1º al 15.
- 5 nombres del 46 al 60 i
- 5 nombres del 61 al 75.

• 5 nombres del 31 al 45.

El procediment retornà un array (que serà el cartró) i la dimensió de l'array. Retornarà l'array ple amb els números ordenats de menor a major.

UTILITZEU (no implementar) els següents procediments o funcions:

- int GenerarNombre (int min, int max, int a[], int dim)
Genera un nombre aleatori entre min i max, i revisa que no estigui a l'array a de dimensió dim.
- void Ordena (int a[], int dim)
Ordena l'array a de dimensió dim de menor a major.

$$200 + 60 = 260$$
$$100 \times 10 = \underline{1000}$$

Resultat: 1200

$$30 \times 40 = 1200$$

Exercici 3 (1 punt)

Fer un procediment anomenat ImprimirBolesSortides per imprimir els nombres de les boles que ja han sortit. El procediment retornà un array i la seva dimensió. L'array contingrà informació sobre si un nombre ha sortit o no. Si el valor d'una determinada posició (índex) és un 1, voldrà dir que la bola corresponent ha sortit i si el és un 0, voldrà dir que aquella bola no ha sortit.

NOTA: Recordeu que els números de les boles van de 1 a 75.

L'operació & l'array
Fer la funció anomenada ArraysIguals per saber si dos arrays tenen tots els valors idèntics. La funció rebrà dos arrays i la seva dimensió. Retornarà true en el cas que siguin iguals i false en cas contrari.
NOTA: Podeu suposar que els dos arrays tenen la mateixa dimensió.

Exercici 4 (1 punt)

Fer una funció anomenada `Escrutini` per saber si un cartó ha aconseguit Bingo. La funció rebrà un array que serà el cartó, la dimensió del cartó i un array (bombo) que contindrà informació dels nombres que han sortit. En l'array bombo si el valor d'una determinada posició (`index`) és un 1, vindrà dir que la bola correspondent ha sortit; si el és un 0, vindrà dir que aquella bola no ha sortit.

La funció retornarà `true` en el cas que s'hagi aconseguit Bingo i `false` en cas contrari.

Exercici 6 (2,5 punts)

Suposeu que teniu els següents procediments i funcions ja implementats:

```
void MenuPrincipal()
{
    cout << " --- Menu Principal --- " << endl;
    cout << "1.- Asignar cartons " << endl;
    cout << "2.- Jugar" << endl;
    cout << "3.- Marcador" << endl;
    cout << "4.- Sortir" << endl;
}
```

Mostra per pantalla un menú amb les opcions del codi: 1. Asignar cartons, 2. Jugar, etc.

voind InicialitzarArray(int a[], int dim, int v)

Inicialitzar el vector a de dimensió dim al valor v

int GenerarNombre(int min, int max, int a[], int dim)

Retorna un enter aleatori entre min i max, controlant que no estigui a l'array a de dimensió dim

Fer un programa complet (declaracions globals i funció main()) que segueixi els següents passos:

1. Declarar (on corresponguï) les constants necessàries per tal que el programa sigui fàcilment modificable.
2. Declarar (on corresponguï) els arrays necessaris per representar els cartons de dos jugadors anomenats `cartoPlayer1` i `cartoPlayer2`. Declarar també un array, anomenat `bombo`, per controlar quins nombres han sortit i quins no.
3. Utilitzar el procediment `MenuPrincipal()` i llegir l'opció escollida.
4. Si l'opció és 1, generar dos cartons (un per a cada jugador) seguint els passos següents:
 - 4.1. Inicialitzar els cartons dels jugadors tot a 0 amb el procediment `InicialitzarArray`.
 - 4.2. Utilitzar el procediment `CrearCarto` per crear els cartons del jugador 1 i del jugador 2.
- 4.3. Comprovar si els dos cartons són diferents utilitzant la funció `arraysIguals`. En cas que siguin iguals tornar a generar un dels dos.
5. Si l'opció és 2, implementa el joc seguint els passos següents:
 - 5.1. S'haurà de comprovar que s'han generat els cartons (s'ha entrat a l'opció 1). En cas contrari, donar un missatge d'error i tornar al menú principal sense fer res més.
 - 5.2. Inicialitzar l'array `bombo` a 0 amb el procediment `InicialitzarArray`.
 - 5.3. Generar un nombre per simular la bola amb la funció `GenerarNombre`.
 - 5.4. Imprimir el nombre que ha sortit i utilitzar la funció `ImprimirBolesSortides` per imprimir els nombres de les boles que han sortit al llarg de la partida.
 - 5.5. Utilitzar la funció `Escrutini` per comprovar si algun dels dos jugadors ha aconseguit fer Bingo.
 - 5.6. En cas que no hi hagi cap guanyador tornar al punt 5.3
6. Si l'opció és 3: imprimir el marcador de les partides que s'han jugat i mostrar el missatge generat pel procediment `ImprimirComentari`.
7. Si l'opció és 4, imprimir el marcador final i mostrar el missatge: "Sortint del programa...".
8. Qualsevol altra opció, escriure el missatge: "Opció no permesa".
9. Repetir els passos 3 a 9, fins que l'opció del menú escollida sigui la 4.

Exercici 7 (0,5 punts)

Declarar un nou tipus de dades, `TCartro`, com un registre amb els camps: `Ident_Cartro` (valor enter), `Numeros` (array de 25 enters) i `Nom_Jugador` (cadena de 50 caràcters).
Fer un procediment anomenat `Imprimir` que rebi com a paràmetre un registre del tipus `TCartro` i imprimeix l'identificador del cartró i el nom del jugador separant-los amb el caràcter '-'.

Exercici 9 (1 punt)

Escriviu la funció `Elevar` que retorna el resultat d'elevat un nombre al quadrat. La funció tindrà dos paràmetres de tipus `int`. El primer serà la base i el segon serà el resultat de l'operació en el cas que es pugui realizar. La funció retornarà un 1 si s'ha pogut realitzar l'operació i un 0 en cas contrari.

NOTA: El valor de la base ha d'estar inclos en l'intervall [-46.340,46.340] per evitar overflow.

Exercici 8 (1 punt)

Fer una funció anomenada `Es_MatPositiva` que rebi com a paràmetres: una matriu d'enters de 4 columnes, el número de files de la matriu, i el número de columnes de la matriu, i retorna 1 si tots els valors de la matriu són positius o zero, i un 0 en cas contrari.

anaments d'Informatàtica (1U3806)

Curs 2020-2021
Examen Parcial (23 de Novembre de 2020)
Nom estudiant: Marcia Rodona Gordeu

Important: Recordeu que cal donar les millors solucions possibles en cada exercici. A més de funcionar correctament, el codi ha d'estar ben programat (codi clar, amb les instruccions més adients, sense operacions ni variables innecessàries, etc.)

Les funcions i procediments de les preguntes de la 1 a la 6 formen part d'un únic programa i, per tant, ha d'haver coherentia entre les seves definicions, i la forma en què s'utilitzen en altres preguntes. El context del programa a desenvolupar, el teniu a continuació:

El Bingo de 75 boles és un joc d'atzar on hi ha 75 boles numerades de l'1 al 75 dins del bombo. Els cartells tenen un total de 25 nombres. En aquest tipus de bingo, es van treure boles del bombo i es guanya un premi quan han sortit tots els nombres del nostre cartell. Això es coneix com a bingo o full House.

Volem fer un programa que permeti jugar al Bingo a dos jugadors. Els jugadors podran jugar tantes parties com vulguin i caldrà portar el compte de quantes parties ha guanyat cadaascú.

Exercici 1 (1 punt)
Fer un procediment anomenat CrearCartell per a generar un cartell. S'han de generar 25 nombres del 1 al 75. S'ha d'assegurar que hi hagin:

- 5 nombres del 1 al 15,
- 5 nombres del 16 al 30,
- 5 nombres del 31 al 45,
- 5 nombres del 46 al 60 i
- 5 nombres del 61 al 75.

El procediment rebrà un array (que serà el cartell) i la dimensió de l'array. Retornarà l'array ple amb els números ordenats de menor a major.

UTILITZEU (no implementar) els següents procediments o funcions:

- void GenerarNombre (int min, int max, int a[], int dim)
Genera un nombre aleatori entre min i max, i revisa que no estigui a l'array a de dimensió dim.
- void Ordena (int a[], int dim)
Ordena l'array a de dimensió dim de menor a major.

$$16 + 15 = 31$$
$$30 - 10 = \underline{\underline{20}}$$

Fer la funció anomenada ArraysIguals per saber si dos arrays tenen tots els valors idèntics. La funció rebrà dos arrays i la seva dimensió. Retornarà True en el cas que siguin iguals i False en cas contrari.

NOTA: Podeu suposar que els dos arrays tenen la mateixa dimensió.

$$100 + 60 = 160$$

$$\text{Resultat} : 1400$$

Exercici 3 (1 punt)

Fer un procediment anomenat ImprimirBolesSortides per imprimir els nombres de les boles que ja han sortit. El procediment rebrà un array i la seva dimensió. L'array contindrà informació sobre si un nombre ha sortit o no. Si el valor d'una determinada posició (index) és un 1, voldrà dir que la bola corresponent ha sortit i si el és un 0, voldrà dir que aquella bola no ha sortit.

NOTA: Recordeu que els números de les boles van de 1 a 75.

Exercici 4 (1 punt)

Fer una funció anomenada Escrutini per saber si un cartró ha aconseguit Bingo. La funció rebrà un array que serà el cartró, la dimensió del cartró i un array (bombo) que contindrà informació dels nombres que han sortit. En l'array bombo si el valor d'una determinada posició (index) és un 1, vindrà dir que la bola correspondent ha sortit i si el és un 0, vindrà dir que aquella bola no ha sortit.

La funció retornarà true en el cas que s'hagi aconseguit Bingo i false en cas contrari.

Exercici 5 (2.5 punts)

Suposeu que teniu els següents procediments i funcions ja implementats:

```
void MenuPrincipal()
{
    cout << " --- Menú Principal --- " << endl;
    cout << "1. - Asignar cartrons " << endl;
    cout << "2. - Jugar " << endl;
    cout << "3. - Marcador " << endl;
    cout << "4. - Sortir " << endl;
}
```

Mostra per pantalla un menú amb les opcions del codi: 1. Assignar cartrons, 2. Jugar, etc.

```
void InicialitzarArray(int a[], int dim, int v)
{
    int GenerarNombre (int min, int max, int a[], int dim)
```

Retorna un enter aleatori entre min i max, controlant que no estigui a l'array a de dimensió dim.

Fer un programa complet (declaracions globals i funció main()) que segueixi els següents passos:

1. Declarar (o correspongui) les constants necessàries per tal que el programa sigui fàcilment modificable.
2. Declarar (o correspongui) els arrays necessaris per representar els cartrons de dos jugadors anomenats cartroPlayer1 i cartroPlayer2. Declarar també un array, anomenat bombo, per controlar quins nombres han sortit i quins no.
3. Utilitzar el procediment MenuPrincipal() i llegir l'opció escollida.
4. Si l'opció és 1, generar dos cartrons (un per a cada jugador) seguint els passos següents:
 - 4.1. Inicialitzar els cartons dels jugadors tot a 0 amb el procediment InicialitzarArray.
 - 4.2. Utilitzar el procediment CrearCartro per crear els cartons del jugador 1 i del jugador 2.
 - 4.3. Comprovar si els dos cartons són diferents utilitzant la funció ArraysIguals. En cas que siguin iguals tornar a generar un dels dos.
5. Si l'opció és 2, implementa el joc seguint els passos següents:
 - 5.1. S'haurà de comprovar que s'han generat els cartrons (s'ha entrat a l'opció 1). En cas contrari, donar un missatge d'error i tornar al menú principal sense fer res més.
 - 5.2. Inicialitzar l'array bombo a 0 amb el procediment InicialitzarArray.
 - 5.3. Generar un nombre per simular la bola amb la funció GenerarNombre.
 - 5.4. Imprimir el nombre que ha sortit i utilitzar la funció ImprimirBolesSortides per imprimir els nombres de les botes que han sortit al llarg de la partida.
 - 5.5. Utilitzar la funció Escrutini per comprovar si algun dels dos jugadors ha aconseguit fer Bingo.
 - 5.6. En cas que no hi hagi cap guanyador tornar al punt 5.3
6. Si l'opció és 3: imprimir el marcador de les partides que s'han jugat i mostrar el missatge generat pel procediment ImprimirComentari.
7. Si l'opció és 4, imprimir el marcador final i mostrar el missatge: "Sortint de ² programa...".
8. Qualsevol altra opció, escriure el missatge: "Opció no permesa".
9. Repetir els passos 3 a 9, fins que l'opció del menú escollida sigui la 4.

Exercici 7 (0.5 punts)

Declarar un nou tipus de dades, `TCartró`, com un registre amb els camps: `Ident_Cartró` (valor enter), `Numeros` (array de 25 enters) i `nom_Jugador` (cadena de 50 caràcters).

Fer un procediment anomenat `Imprimir` que rebi com a paràmetre un registre del tipus `TCartró` i imprimeix l'identificador del cartró i el nom del jugador separant-los amb el caràcter `:`.

Exercici 9 (1 punt)

Escriure la funció `Elevar` que retorni el resultat d'elevat un nombre al quadrat. La funció tindrà paràmetres de tipus `int`. El primer serà la base i el segon serà el resultat de l'operació en el cas que es pugui realitzar. La funció retornarà un 1 si s'ha pogut realitzar l'operació i un 0 en cas contrari.

NOTA: El valor de la base ha d'estar inclòs en l'intervall [-46.340, 46.340] per evitar overflow.

Exercici 8 (1 punt)

Fer una funció anomenada `Es_MatPositiva` que rebi com a paràmetres una matriu d'enters de 4 columnes, el número de files de la matriu, i el número de columnes de la matriu, i retorna 1 si tots els valors de la matriu són positius o zero, i un 0 en cas contrari.

Exercicis d'Informatàtica (103806)

Curs 2020-2021

Examen Parcial (23 de Novembre de 2020)
Nom estudiant: Víctor Josep Núñez

Grup: NIU: 4216242

Exercici 2 (1 punt)
Fer la funció anomenada `ArraysIguals` per saber si dos arrays tenen tots els valors idèntics. La funció rebrà dos arrays i la seva dimensió. Retornarà `True` en el cas que siguin iguals i `False` en cas contrari.

NOTA: Podeu suposar que els dos arrays tenen la mateixa dimensió.

Important: Recordeu que cal donar les millors solucions possibles en cada exercici. A més de funcionar correctament, el codi ha d'estar ben programat (codi clar, amb les instruccions més adients, sense operacions ni variables innecessàries, etc.) Les funcions i procediments de les preguntes de la 1 a la 6 formen part d'un únic programa i, per tant, ha d'haver coherentia entre les seves definicions, i la forma en què s'utilitzen en altres preguntes. El context del programa a desenvolupar, el tenui a continuació:

El Bingo de 75 boles és un joc d'atzar on hi ha 75 boles numerades de l'1 al 75 dins del bombo.

Els cartons tenen un total de 25 nombres. En aquest tipus de bingo, es van treient boles del bombo i es guanya un premi quan han sortit tots els nombres del nostre cartó. Això es coneix com a bingo o full House.

Volem fer un programa que permeti jugar al Bingo a dos jugadors. Els jugadors podran jugar tantes partides com vulguin i caindrà portar el compte de quantes partides ha guanyat cadascú.

Exercici 1 (1 punt)

Fer un procediment anomenat `CrearCartro` per a generar un cartó. S'han de generar 25 nombres del 1 al 75. S'ha d'assegurar que hi hagin:

- 5 nombres del 1 al 15,
- 5 nombres del 16 al 30,
- 5 nombres del 31 al 45,
- 5 nombres del 46 al 60 i
- 5 nombres del 61 al 75.

El procediment rebrà un array (que serà el cartó) i la dimensió de l'array. Retornarà l'array ple amb els números ordenats de menor a major.

UTILITZEU (no implementar) els següents procediments o funcions:

- `int GenerarNombre (int min, int max, int arr[], int dim)`
- `void Ordena (int a[], int dim)`

Ordene l'array a de dimensió dim de menor a major.

$$600 + 40 = \underline{\underline{640}}$$
$$300 - 5 = \underline{\underline{295}}$$

$$23 + 23 = 46$$
$$50 \times 2 = 100$$

Resultat: 100

Exercici 3 (1 punt)

Fer un procediment anomenat `ImprimirBolesSortides` per imprimir els nombres de les boles que ja han sortit. El procediment rebrà un array i la seva dimensió. L'array contingdrà informació sobre si un nombre ha sortit o no. Si el valor d'una determinada posició (`index`) és un 1, voldrà dir que la bola corresponent ha sortit i si el és un 0, voldrà dir que aquella bola no ha sortit.

NOTA. Recordeu que els números de les boles van de 1 a 75.

Exercici 4 (1 punt)

Fer una funció anomenada `Escrutini` per saber si un cartró ha aconseguit Bingo. La funció rebrà un array que serà el cartró, la dimensió del cartró i un array (bombo) que contingrà informació dels nombres que han sortit. En l'array bombó si el valor d'una determinada posició (`index`) és un 1, vindrà dir que la bola corresponent ha sortit i si el és un 0, vindrà dir que aquella bola no ha sortit.

La funció retornarà `True` en el cas que s'hagi aconseguit Bingo i `False` en cas contrari.

Exercici 6 (2.5 punts)

Suposeu que teniu els següents procediments i funcions ja implementats:

```
void MenúPrincipal()
{
    cout << "---- Menú Principal ----" << endl;
    cout << "1.- Asignar cartrons" << endl;
    cout << "2.- Jugar" << endl;
    cout << "3.- Marcador" << endl;
    cout << "4.- Sortir" << endl;
}
```

Mostra per pantalla un menú amb les opcions del codi: 1. Asignar cartrons, 2. Jugar, etc.

```
void InicialitzarArray(int a[], int dim, int v)
```

Inicialitzar el vector a de dimensió dim al valor v

int GeneraNombre(int min, int max, int a[], int dim)

Retorna un enter aleatori entre min i max, controlant que no estigui a l'array a de dimensió dim

Fer un programa complet (declaracions globals i funció main()) que segueixi els següents passos:

1. Declarar (on correspongui) les constants necessàries per tal que el programa sigui fàcilment modificable.
2. Declarar (on correspongui) els arrays necessaris per representar els cartrons de dos jugadors anomenats `cartroPlayer1` i `cartroPlayer2`. Declarar també un array, anomenat `bombo`, per controlar quins noms han sortit i quins no.
3. Utilitzar el procediment `MenúPrincipal()` i llegir l'opció escollida.
4. Si l'opció és 1, generar dos cartrons (un per a cada jugador) seguint els passos següents:
 - 4.1. Inicialitzar els cartrons dels jugadors tot a 0 amb el procediment `InicialitzarArray`.
 - 4.2. Utilitzar el procediment `CrearCartro` per crear els cartrons del jugador 1 i del jugador 2.
 - 4.3. Comprovar si els dos cartrons són diferents utilitzant la funció `ArraysEquals`. En cas que siguin iguals tornar a generar un dels dos.
5. Si l'opció és 2, implementa el joc seguint els passos següents:
 - 5.1. S'haurà de comprovar que s'han generat els cartrons (s'ha entrat a l'opció 1). En cas contrari, donar un missatge d'error i tornar al menú principal sense fer res més.
 - 5.2. Inicialitzar l'array `bombo` a 0 amb el procediment `InicialitzarArray`.
 - 5.3. Generar un nombre per simular la bola amb la funció `GeneraNombre`.
 - 5.4. Imprimir el nombre que ha sortit i utilitzar la funció `ImprimirBolesSortides` per imprimir els nombres de les boles que han sortit al llarg de la partida.
 - 5.5. Utilitzar la funció `Escrutini` per comprovar si algun dels dos jugadors ha aconseguit fer Bingo.
 - 5.6. En cas que no hi hagi cap guanyador tornar al punt 5.3
6. Si l'opció és 3: imprimir el marcador de les partides que s'han jugat i mostrar el missatge generat pel procediment `ImprimirComentari`.
7. Si l'opció és 4, imprimir el marcador final i mostrar el missatge: "Sortint del programa...".
8. Qualsevol altra opció, escriure el missatge: "Opció no permesa".
9. Repetir els passos 3 a 9, fins que l'opció del menú escollida sigui la 4.

Exercici 7 (0.5 punts)

Declarar un nou tipus de dades, `TCartro`, com un registre amb els camps: `Ident_Cartro` (valor ente), `Numeros` (array de 25 enters) i `Non_Jugador` (cadena de 50 caràcters).

Fer un procediment anomenat `Imprimir` que rebi com a paràmetre un registre del tipus `TCartro` i imprimeix l'identificador del cartró i el nom del jugador separant-los amb el caràcter `,`.

Exercici 9 (1 punt)

Escríure la funció `Elevar` que retorna el resultat de elevar un nombre al quadrat. La funció tindrà paràmetres de tipus `int`. El primer serà la base i el segon serà el resultat de l'operació en el cas que es pugui realizar. La funció retornarà un 1 si s'ha pogut realitzar l'operació i un 0 en cas contrari.

NOTA: El valor de la base ha d'estar inclòs en l'intervall [-46.340,46.340] per evitar overflow.

Exercici 8 (1 punt)

Fer una funció anomenada `Es_MatPositiva` que rebi com a paràmetres: una matriu d'enters de 4 columnes, el número de files de la matriu, i el número de columnes de la matriu, i retorna 1 si tots els valors de la matriu són positius o zero, i un 0 en cas contrari.