

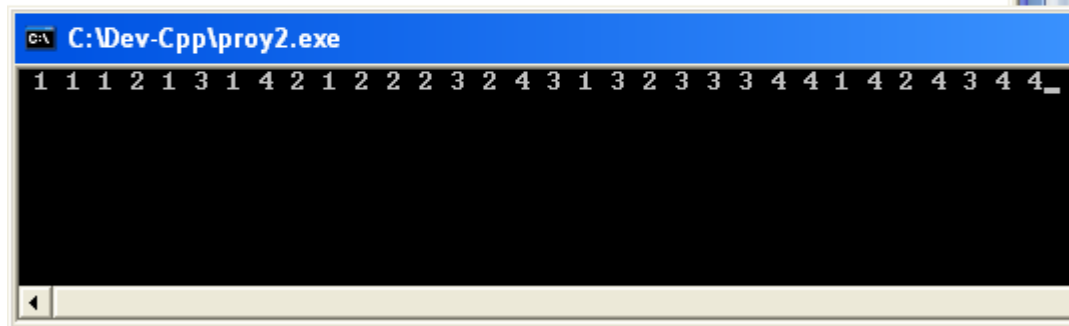
For anidados

Parte 1

Un for (i=1;i<5;i++) nos dà 1,2,3,4

Si ponemos dos utilizando ahora además de la variable i la variable j. Tenemos

```
main()
{
  int i,j;
  for (i=1;i<5;i++)
    for (j=1;j<5;j++)
      printf(" %d %d", i, j);
```



```
getch();
}
```

Pues aparentemente se ven números sin un orden dado, pero ya sabemos que

Si con un for me dan un grupo de números en orden ascendente (o descendente), con dos for's me daría dos grupo de números en orden ascendente (o descendente).

Su aplicación inmediata es para mostrar una matriz

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1j} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2j} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{i1} & a_{i2} & \dots & a_{ij} & \dots & a_{in} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mj} & \dots & a_{mn} \end{pmatrix}$$

Acomodemos para verlo asi. Únicamente con un printf("\n"); intercalado. Observe que

A_{ij} $i=1,2,\dots,n$ $j=1,2,\dots,n \Rightarrow a_{12}$ tiene $i=1, j=2$; a_{22} tiene $i=2, j=2$;

Y la letra i representara nuestros renglones y j nuestras columnas.

Cuando i cambie de 1 a 2 pondremos el printf, para que se separe y se vea como la matriz A.

Parte 2.Observe y mencione en su tarea donde se escribió el \n de los siguientes ejemplos.

Escriba y corra los ejemplos.

a)

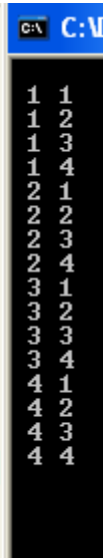
```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>

main()
{
    int i,j;

    for (i=1;i<5;i++)

        for (j=1;j<5;j++)
            printf("\n %d %d",i, j);

    getch();
}
```



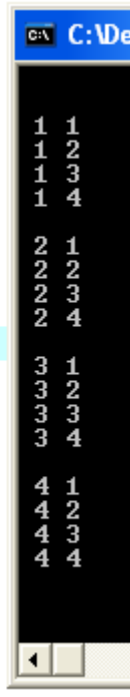
b)

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>

main()
{
    int i,j;

    for (i=1;i<5;i++)
    {
        printf("\n ");
        for (j=1;j<5;j++)
            printf("\n %d %d",i, j);
    }

    getch();
}
```



c)

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
```

```
main()
{
    int i,j;

    for (i=1;i<5;i++)
    {
        printf("\n ");
        for (j=1;j<5;j++)
            printf(" %d %d",i, j);

    }

    getch();
}
```



d)

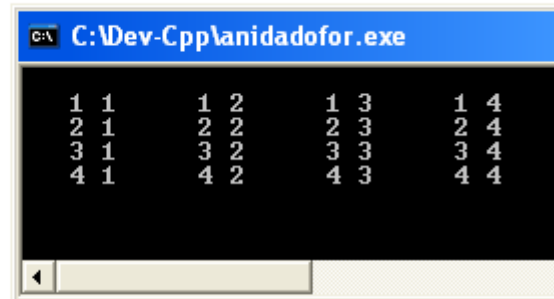
```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
```

```
main()
{
    int i,j;

    for (i=1;i<5;i++)
    {
        printf("\n ");
        for (j=1;j<5;j++)
            printf(" %d %d  ",i, j);

    }

    getch();
}
```



Parte 3. Recuerdan que una matriz transpuesta es así.

$$A = \begin{bmatrix} 10 & 20 & 30 \\ 40 & 50 & 60 \end{bmatrix} \quad A' = \begin{bmatrix} 10 & 40 \\ 20 & 50 \\ 30 & 60 \end{bmatrix}$$

Nosotros aun no hacemos ninguna operación solo estamos jugando con los índices. Para mostrar una matriz transpuesta en vez de mostrar i con j; lo invertimos j con i.

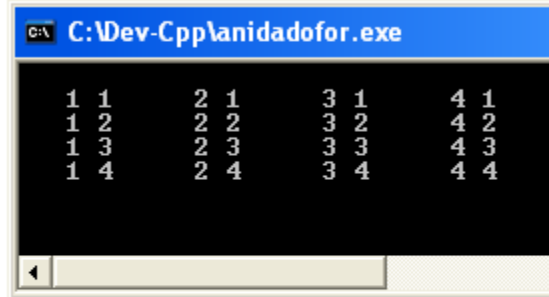
```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
```

```
main()
{
  int i,j;

  for(i=1;i<5;i++)
  {
    printf("\n ");
    for(j=1;j<5;j++)
      printf(" %d %d ",j, i);

  }

  getch();
}
```



Obsérvese que lo único que se cambio fue el printf final antes era i,j y ahora se puso j,i.

Parte 4.

Dejemos un for solo para ver que hacen.

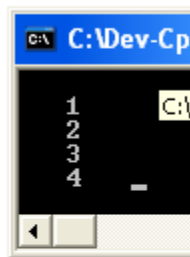
```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
```

```
main()
{
  int i,j;

  for(i=1;i<5;i++)
  {
    printf("\n ");
    // for(j=1;j<5;j++)
    // printf(" %d %d ",j, i);
    printf(" %d ", i);

  }

  getch();
}
```



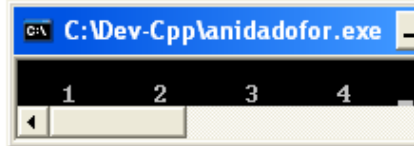
```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
```

```
main()
{
  int i,j;

  //for(i=1;i<5;i++)
  //{
    printf("\n ");
    for(j=1;j<5;j++)
      printf(" %d %d ",j, i);
    printf(" %d ", j);

  // }

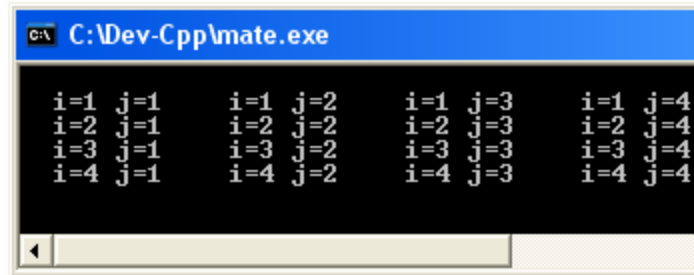
  getch();
}
```



Ambos imprimen números del 1 al 4, pero juntos, primero hace el interno y luego el externo.

Si pudiéramos mas SIEMPRE haría el interno primero.

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <math.h>
int i,j;
int main()
{
    for(i=1;i<5;i++)
    {
        printf("\n");
        for(j=1;j<5;j++)
            printf(" i=%d j=%d ", i,j);
    }
}
```



```
getch();
return 0;
}
```

O sea que primero imprime

j=1 y luego combina las j=1,2,3,4

i=2 j=1,2,3,4

i=3 j=1,2,3,4

i=4 j=1,2,3,4

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
```

```
main()
{
    int i,j;

    for(i=1;i<5;i++)

        for(j=1;j<5;j++)
            printf("\n %d %d",i, j);
}
```

```
getch();
}
```



Parte 5 . Para tres for anidados, restringiremos a $i < 3$, para que se vea mejor la idea.

```

int i,j,k;
int main()
{

    for(i=1;i<3;i++)
    {
        //printf("\n");
        for(j=1;j<3;j++)
            for(k=1;k<3;k++)
                printf(" %d %d %d ", i,j, k);
    }
}

```

Note que ahora hace

	A	B	C
1	i=1	j=1	
2			k=1
3			k=2
4		j=2	
5			k=1
6			k=2
7	i=2	j=1	
8			k=1
9			k=2
10		j=2	
11			k=1
12			k=2

O sea ejecuta primero las ks, luego lo que quedo en las js y finalmente continua con lo que deajo en las is. (Los internos termina el ciclo completo y se va terminando hacia el exterior)

Vamos a intentar jugar con estos índices.

Parte 6.

- a) Imprime estos índices.

Primero inténtalo y luego consulta la respuesta programa.

```

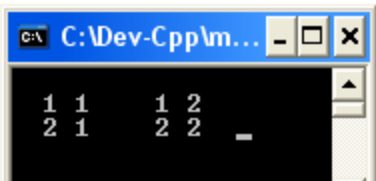
int i,j,k;
int main()
{
    for (i=1;i<3;i++)
    {
        printf("\n");
        for (j=1;j<3;j++)

            printf("  %d %d  ", i,j);

    }

    getch();
}

```



b) Imprime estos índices

```

1 1    1 2    1 3
2 1    2 2    2 3

```

c)

```

1 1
2 1
3 1
4 1

```

```

1 1    2 1    3 1    4 1

```

d)

Resp b)

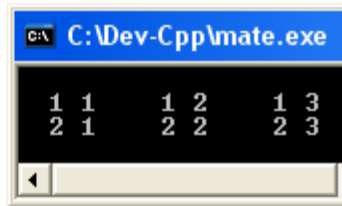
```

int i,j,k;
int main()
{
    for (i=1;i<3;i++)
    {
        printf("\n");
        for (j=1;j<4;j++)

            printf("  %d %d  ", i,j);

    }
}

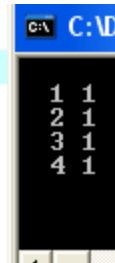
```



Resp c)

```
for (i=1; i<5; i++)
{
    printf("\n");
    for (j=1; j<2; j++)

        printf(" %d %d ", i, j);
```

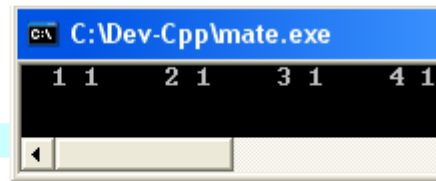


```
C:\ C:\D
1 1
2 1
3 1
4 1
```

Resp d)

```
for (i=1; i<5; i++)
{
    //printf("\n");
    for (j=1; j<2; j++)

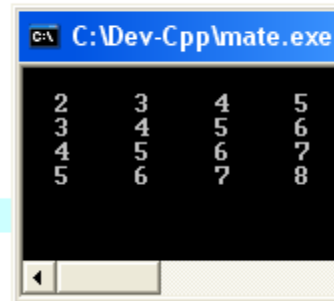
        printf(" %d %d ", i, j);
```



```
C:\ C:\Dev-Cpp\mate.exe
1 1 2 1 3 1 4 1
```

Parte 7. Ejemplo. Sumando sus índices.

```
int i, j, k;
int main()
{
    for (i=1; i<5; i++)
    {
        printf("\n");
        for (j=1; j<5; j++)
            printf(" %d ", i+j);
    }
```



```
C:\ C:\Dev-Cpp\mate.exe
2 3 4 5
3 4 5 6
4 5 6 7
5 6 7 8
```

Parte 8. Grupo de Registros guardados con un nombre común más dos índices.

Recordando que tenemos casilleros de memoria llamados a gusto de nosotros por ejemplo A, B, C, de tipo entero o doble.

A	B	C
---	---	---

También podemos tener casilleros de memoria cuadrículares o por renglón (o columna) pero llamados con un nombre en común. Por ejemplo. Todo un grupo llamado A u otro grupo llamado B y cada casillero del grupo puede escogerse con el grupo de índices.

B	
11	12
21	22

A	
11	12
21	22

Por ejemplo quiero guardar un valor en el casillero A[2][1] por ejemplo el 5, podríamos escribirlo así: A[2][1]=5. (Deben escribirse entre paréntesis cuadrado, los índices),

B	
11	12
21	22

A	
11	12
21 5	22

Y si quiero mandarlo a imprimir

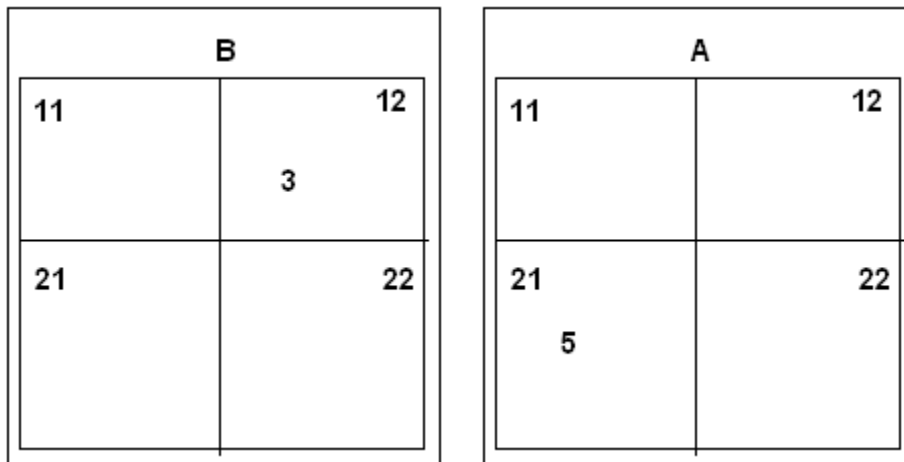
`printf("%d",A[2][1]);` y por supuesto imprime el 5.

Si quisiéramos guardar por teclado en el B[1][2] el valor de 3.

Quedaría así:

`scanf("%d",&B[1][2]);`

Le damos el valor de 3 y enter. Luego mandamos a imprimir `printf("%d",B[1][2]);`



Y trabajar los índices en grupos de registro es muy divertido.

Parte 9. Cópialo y correlo

```
#include <stdio.h>
#include <conio.h>

int i,j,a[20][20];
int main()
{
    for (i=1;i<3;i++)
    {
        // printf("\n");
        for (j=1;j<3;j++)
        {
            printf("\ndame el valor de %d%d:",i,j);
            scanf("%d", &a[i][j]);
            printf(" valor obtenido%d \n ", a[i][j]);
        }
    }

    getch();
    return 0;
}
```

```
C:\Dev-Cpp\mate.exe
dame el valor de 11:1
valor obtenido1
dame el valor de 12:2
valor obtenido2
dame el valor de 21:3
valor obtenido3
dame el valor de 22:4
valor obtenido4
```

Checando que siguen ahí los números guardados

```

int i,j,a[20][20],suma[20][20];
int main()
{
    for(i=1;i<3;i++)
    {
        // printf("\n");
        for(j=1;j<3;j++)
        {
            printf("\ndame el valor de %d%d:",i,j);
            scanf("%d", &a[i][j]);
            printf(" valor obtenido%d \n ", a[i][j]);
        }
    }
    //checando que siguen ahi los numeros

    for(i=1;i<3;i++)
        for(j=1;j<3;j++)
            printf(" %d  ", a[i][j]);

    getch();
    return 0;
}

```

C:\Dev-Cpp\mate.exe

```

dame el valor de 11:1
valor obtenido1

dame el valor de 12:2
valor obtenido2

dame el valor de 21:3
valor obtenido3

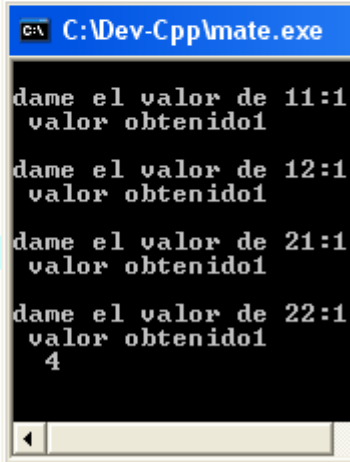
dame el valor de 22:4
valor obtenido4
1 2 3 4

```

Parte 10. Ejemplo: suma los valores del grupo de registros de "a".

```
int i,j,a[20][20],suma;
int main()
{
    for(i=1;i<3;i++)
    {
        // printf("\n");
        for(j=1;j<3;j++)
        {
            printf("\ndame el valor de %d%d:",i,j);
            scanf("%d", &a[i][j]);
            printf(" valor obtenido%d \n ", a[i][j]);
        }
    }
    //checando que siguen ahí los números
    suma=0;
    for(i=1;i<3;i++)
        for(j=1;j<3;j++)
            suma=suma+a[i][j];
    printf(" %d ", suma);

    getch();
    return 0;
}
```



```
C:\Dev-Cpp\mate.exe
dame el valor de 11:1
valor obtenido1
dame el valor de 12:1
valor obtenido1
dame el valor de 21:1
valor obtenido1
dame el valor de 22:1
valor obtenido1
4
```

NOTA: para declarar las variables de grupos de registros deben ser SIEMPRE MUCHO MAYOR a los utilizados, pues si pones un valor menor al que utilizaras se pueden borrar partes de la memoria de TU computadora.

Esto quiere decir: en los for utilizamos de 1 a <3 o sea 1,2 Entonces estamos trabajando con DOS registros de memoria en i y también dos registros de memoria en j, y en las declaraciones apartamos 20 para i y 20 para j

```
int a[20][20] => for(i=1;i<3;i++)
                 for(j=1;j<3;j++)
```

PERO estaría mal si hubiese apartado menos memoria: int a[1][1]; CUIDADO!!!,

RECOMENDACIÓN: SI ESTAS EMPEZANDO A PROGRAMAR APARTA MEMORIA MUCHO MAS QUE LO QUE UTILIZARAS, DESPUES SEREMOS MAS PRECISOS.

Ejercicios:

- 1) CREA UN GRUPO DE REGISTRO A[][] DE 2X2 COMO EL EJEMPLO ANTERIOR Y OTRO DE B[][] TAMBIEN DE 2X2. RESTA SUS VALORES CONTENIDOS.

2) A ESTOS GRUPOS DE REGISTROS ESPECIALES LES LLAMAREMOS ARREGLOS. CREAR UN ARREGLO DE 3X3 Y MANDA A PEDIR LOS DATOS DEL TECLADO Y MANDALOS A IMPRIMIR.