

UTN HAEDO

Trabajo Práctico de Computación N°2

Curso: 1°1ª y 1°6ª

Horario de cursada: sábados de 11 a 12⁴⁰

Integrantes:

- Ambertine Alan
- Deon Luciano
- Sánchez Pablo Hernán

Nota:

AÑO 2015

- **Generalidades:**

El mismo puede realizarse en grupo de hasta cuatro integrantes o de forma Individual.

La presentación del mismo debe ser formal, presentando Diagrama y Código impresos, teniendo disponible el código en forma digital para realizar una ejecución del mismo.

Todas las hojas presentadas deben tener los datos de quienes realizaron el TP.

Los alumnos integrantes del grupo deben explicar los criterios utilizados en el desarrollo del TP

- **Enunciado**

Ejercicio 1:

1. Crear una función llamada **producto** que reciba dos números enteros para trabajar y devuelva el producto de ambos números **por sumas sucesivas**.

NO utilizar el operador de la multiplicación (*)

2. Crear una función llamada **división_entera** que reciba para trabajar dos números enteros y devuelva el cociente entero de la división entre el primer número dividido el segundo y el resto de la misma división, obteniendo ambos resultados **por restas sucesivas**.

NO utilizar los operadores de la división entera (/ y %)

El programa principal pide ingresar un conjunto de n pares de números enteros.

De cada par de números ingresado y mediante la conveniente invocación a las funciones **producto** y

division_entera, debe informar:

1. El producto de ambos números
2. La división entera del primer número dividido por el segundo
3. El resto de esa división entera

Nota: Tener en cuenta que el segundo número puede ser cero, en ese caso informar "Error de división por cero".

Ejercicio 2:

Se ingresan: número de historia clínica, valor de la cuota prepaga, cantidad de cuotas vencidas (sin pagar), de un conjunto de n asociados de una clínica, suponer un máximo de 500 asociados.

Con esta información, almacenada en vectores, se pide realizar las siguientes tareas:

1. A medida que se ingresan los datos de cada historia clínica, generar el vector con el importe adeudado de cada historia clínica.
2. A fin de procesar el pago de cuotas vencidas, ingresar un número de historia clínica y la cantidad de cuotas a pagar, buscar el número de historia clínica ingresado en el vector de números de historias clínicas y una vez ubicado, actualizar la cantidad de cuotas vencidas sin pagar y el importe total adeudado que le corresponden a ese número de historia clínica de acuerdo a las cuotas que va a pagar.
Este proceso finaliza al ingresar número de historia clínica 0 (inexistente). Mostrar los vectores actualizados.
3. A partir de los vectores de números de historias clínicas y cuotas vencidas, generar un nuevo vector con los números de historia clínica que adeudan más de 5 cuotas. Mostrar en pantalla el vector generado (si existe).
4. Ordenar el vector de números de historias clínicas por número de menor a mayor con arrastre de los vectores de cantidad de cuotas vencidas, de valor de la cuota y de total de importe adeudado. Mostrar los vectores luego del ordenamiento.

Ejercicio 1

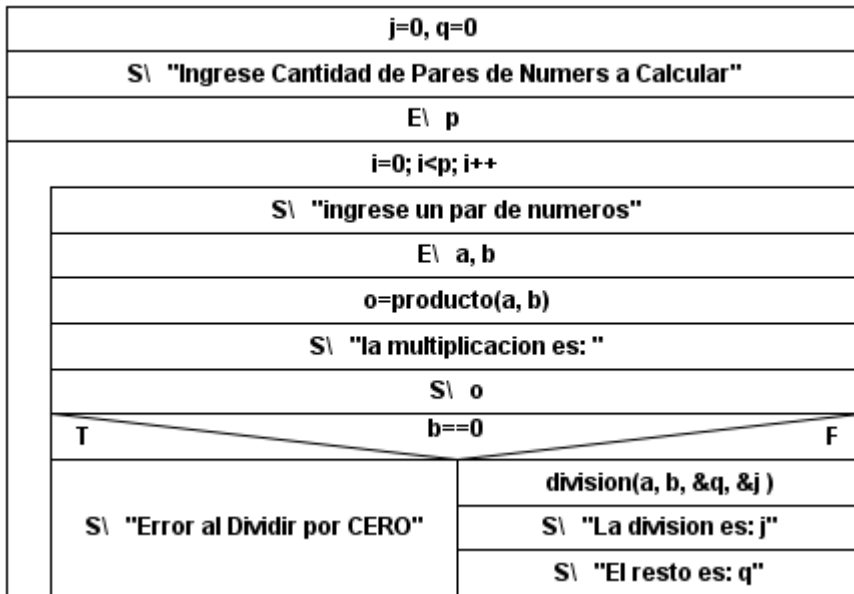
Función producto:

T					z==0 t==0					F														
z=0					T					t>0					F									
					m=z					T					z<0					F				
										z=m					m=z									
										n=t					z.=(m+m)									
					i=0;i<t-1;i++					n.=(t+t)					m=z									
					z+=m					t=n					n=t									
										i=0;i<t-1;i++					n.=(t+t)									
										z+=m					t=n									
										m=z					i=0;i<t-1;i++									
										z.=(m+m)					z+=m									
return z																								

Función división:

b=0, r=0, k=0, flag=0, v=0								
T		x<0&&y<0			F			
aux=x		T		x>0&&y<0		F		
x.=(aux+aux)		aux=y		T		x<0&&y>0		F
aux=y				aux=x				
y.=(aux+aux)		x.=(aux+aux)						
flag=1		v=1		v=1				
		flag=1						
T		x==y			F			
k=1		T		x<y			F	
		k=0		b!=1				
r=0								
		x-=y						
k++								
r=0		T		x<y		F		
				r=x		T		x==y
		k++						
		b=1		r=0		b=1		
T		v==1			F			
aux=k		'j=k						
k.=(aux+aux)								
'j=k								
T		flag==1			F			
aux=r		'z=r						
r.=(aux+aux)								
'z=r								

Diagrama del programa principal:



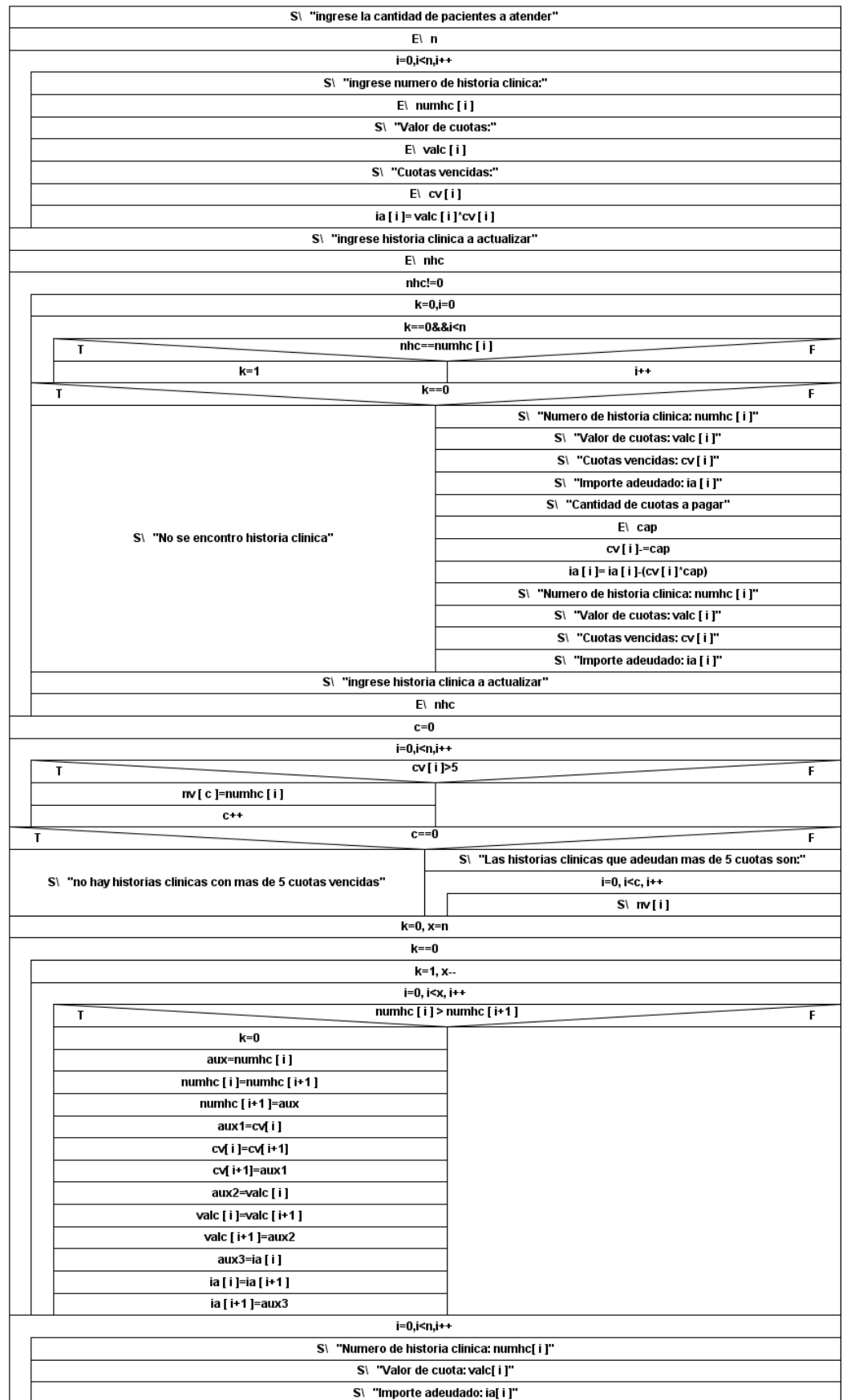
La codificación:

```
#include<stdio.h>
int producto(int z, int t) {
    int m,i,n;
    if (z==0 || t==0)
        z=0;
    else{
        if (t>0) {
            m=z;
            for(i=0; i<t-1; i++) {
                z+=m; } }
        else {
            if (z>0) {
                m=z; n=t; n--=(t+t); t=n;
                for(i=0; i<t-1; i++) { z+=m; }
                m=z; z--=(m+m); }
            else{ m=z; z--=(m+m); m=z; n=t; n--=(t+t); t=n;
                for(i=0; i<t-1; i++) { z+=m; } } } }
    return z; }
void division (int x, int y , int *z , int *j){
    int b=0,r=0,k=0,v=0,flag=0,aux;
    if (x<0&&y<0) {
        aux=x;x--=(aux+aux);
        aux=y;y--=(aux+aux);
        flag=1; }
    else {
        if (x>0&&y<0) {
            aux=y;y--=(aux+aux); v=1; }
        else{
            if (x<0&&y>0) {
                aux=x;x--=(aux+aux); v=1; flag=1; }
            else; } }
    if(x==y) {k=1;r=0; }
    else
```

```
{if (x<y) {k=0;r=x;}
else{
    while (b!=1)
        {x-=y; k++;
        if (x<y) {r=x;b=1;}
        else{
            if (x==y) {k++;r=0;b=1;}
            else;}}}}
if (v==1) {aux=k;k-=(aux+aux);*j=k;}
else {*j=k;}
if (flag==1) {aux=r;r-=(aux+aux);*z=r;}
else *z=r;}
main(){
int p,i,a,b,o,j,q;
j=0;q=0;
printf("Ingrese Cantidad de Pares de Numeros a Calcular\n");
scanf("%d",&p);
for (i=0;i<p;i++){
    printf("ingrese un par de numeros\n");
    scanf("%d%d",&a,&b);
    o=producto(a,b);
    printf("La multiplicacion es:%d\n",o);
    if(b==0)
        printf("Error al Dividir por CERO\n");
    else {division(a,b,&q,&j);
        printf("La Division es:%d\nEl resto es:%d\n",j,q);}
}
system("pause");}
```

Ejercicio 2

Diagrama del programa:



Codificación del programa:

```
#include<stdio.h>
main() {
    int n,i,numhc[500],cv[500],k,nhc,cap,x,c,aux,aux1,nv[500],g;
    float valc[500],ia[500],aux3,aux2,j,f;
    printf("ingrese la cantidad de pacientes a
    atender\n"); scanf("%d",&n);
    for (i=0;i<n;i++){
        printf("\nIngrese:\nNumero de Historia
        clinica:\t"); scanf("%d",&numhc[i]);
        printf("\nValor de Cuota:\t"); scanf("%f",&valc[i]);
        printf("\nCoutas vencidas:\t"); scanf("%d",&cv[i]);
        ia[i]=valc[i]*cv[i];
        printf("\nImporte adeudado:\t%f\n",ia[i]);}
    printf("\nIngrese Historia Clinica a
    Actualizar:\t"); scanf("%d",&nhc);
    while(nhc!=0) {
        k=0;i=0;
        while (k==0&&i<n) {
            if (nhc==numhc[i]) k=1;
            else i++;}
        if (k==0) {
            printf("\nNo se encontro Historia Clinica");}
        else {
            printf("\nNumero de Historia clinica:\t%d",numhc[i]);
            printf("\nValor de Cuota:\t%f",valc[i]);
            printf("\nCoutas vencidas:\t%d",cv[i]);
            printf("\nImporte adeudado:\t%f",ia[i]);
            printf("\nCantidad de cuotas a pagar:\t");
            scanf("%d",&cap);
            cv[i]-=cap;
            j=valc[i];
            f=cap;
            ia[i]-=j*f;
            system("cls");
            printf("\nNumero de Historia clinica:\t%d",numhc[i]);
            printf("\nValor de Cuota:\t%f",valc[i]);
            printf("\nCoutas vencidas:\t%d",cv[i]);
            printf("\nImporte adeudado:\t%f\n",ia[i]);
            system("pause");
            system("cls");}
        printf("\nIngrese Historia Clinica a Actualizar:\n");
        scanf("%d",&nhc);}
    c=0;
    for (i=0;i<n;i++){
        if (cv[i]>5)
            {nv[c]=numhc[i];c++;}
        else;}
    if (c==0)
        printf("No hay Historias Clinicas con mas de 5 cuotas
        vencidas\n");
    else{printf("Las Historias Clinicas con mas de 5 cuotas vencidas
    son:\n");
        for (i=0;i<c;i++) printf("%d\n",nv[i]);}
```



```
k=0;x=n;
while (k==0) {
    k=1;x--;
    for (i=0;i<x;i++) {
        if (numhc[i]>numhc[i+1]) {
            k=0;
            aux=numhc[i];numhc[i]=numhc[i+1];numhc[i+1]=aux;
            aux1=cv[i];cv[i]=cv[i+1];cv[i+1]=aux1;
            aux2=valc[i];valc[i]=valc[i+1];valc[i+1]=aux2;
            aux3=ia[i];ia[i]=ia[i+1];ia[i+1]=aux3;}
        ;}}
    for(i=0;i<n;i++){printf("\nNumero de Historia
clinica:\t%d",numhc[i]);
        printf("\nValor de Cuota:\t%f",valc[i]);
        printf("\nCoutas vencidas:\t%d",cv[i]);
        printf("\nImporte adeudado:\t%f\n",ia[i]);}
system("pause");}
```