

## Introducción a la Programación I – Práctico 1 (Pilas)

```
Program e1p1;

Uses estructu;

Var
  dada,cjto1,cjto2:pila;

Begin
  Readpila(dada);
  Inicpila(cjto1, '');
  Inicpila(cjto2, '');
  if not pilavacia (dada) then
    apilar(cjto1, desapilar(dada));
  if not pilavacia (dada) then
    apilar(cjto1, desapilar(dada));
  if not pilavacia (dada) then
    apilar(cjto1, desapilar(dada));
  if not pilavacia (dada) then
    apilar(cjto2, desapilar(dada));
  if not pilavacia (dada) then
    apilar(cjto2, desapilar(dada));
  Writepila(dada);
  Writepila(cjto1);
  Writepila(cjto2);

End.

/*****/

Program e2p1;

Uses estructu;

Var
  origen,destino:pila;

Begin
  Readpila(origen);
  Inicpila(destino, '');
  While not pilavacia (origen) do
    apilar(destino, desapilar(origen));
  Writepila(origen);
  Writepila(destino);

End.

/*****/
```

```
Program e3p1;

Uses estructu;

Var
  dada, anteriores:pila;

Begin
  Inicpila(dada, '2 3 8 4 2 10');
  Inicpila(anteriores, '');
  While not (tope(dada)=8) do
    apilar(anteriores, desapilar(dada));
  Writepila(dada);
  Writepila(anteriores);

End.

/*****/

Program e4p1;

Uses estructu;

Var
  origen, destino, aux:pila;

Begin
  Readpila(origen);
  Inicpila(destino, '');
  Inicpila(aux, '');
  while not pilavacia (origen) do
    apilar(aux, desapilar(origen));
  while not pilavacia (aux) do
    apilar(destino, desapilar(aux));
  Writepila(origen);
  Writepila(destino);
  Writepila(aux);

End.

/*****/
```

```
Program e5p1;
```

```
Uses estructu;
```

```
Var
```

```
  dada, aux1, aux2:pila;
```

```
Begin
```

```
  Readpila(dada);
```

```
  Inicpila(aux1, '');
```

```
  Inicpila(aux2, '');
```

```
  while not pilavacia (dada) do
```

```
    apilar(aux1, desapilar(dada));
```

```
  while not pilavacia (aux1) do
```

```
    apilar(aux2, desapilar(aux1));
```

```
  while not pilavacia (aux2) do
```

```
    apilar(dada, desapilar(aux2));
```

```
  Writepila(dada);
```

```
  Writepila(aux1);
```

```
  Writepila(aux2);
```

```
End.
```

```
/*-----*/
```

```
Program e6p1;
```

```
Uses estructu;
```

```
Var
```

```
  dada, aux1, aux2:pila;
```

```
Begin
```

```
  Readpila(dada);
```

```
  Inicpila(aux1, '');
```

```
  Inicpila(aux2, '');
```

```
  if not pilavacia (dada) then
```

```
    apilar(aux1, desapilar(dada));
```

```
  while not pilavacia (dada) do
```

```
    apilar(aux2, desapilar(dada));
```

```
  if not pilavacia (aux1) then
```

```
    apilar(dada, desapilar(aux1));
```

```
  while not pilavacia (aux2) do
```

```
    apilar(dada, desapilar(aux2));
```

```
  Writepila(dada);
```

```
  Writepila(aux1);
```

```
  Writepila(aux2);
```

```
End.
```

```
/*-----*/
```

```
Program e7p1;

Uses estructu;

Var
  dada, aux1, aux2:pila;

Begin
  Readpila(dada);
  Inicpila(aux1, '');
  Inicpila(aux2, '');
  while not pilavacia (dada) do
    apilar(aux1, desapilar(dada));
  if not pilavacia (aux1) then
    apilar(aux2, desapilar(aux1));
  while not pilavacia (aux1) do
    apilar(dada, desapilar(aux1));
  if not pilavacia (aux2) then
    apilar(dada, desapilar(aux2));
  Writepila(dada);
  Writepila(aux1);
  Writepila(aux2);

End.

/*****/

Program e8p1;

Uses estructu;

Var
  pozo, jugador1, jugador2:pila;

Begin
  Readpila(pozo);
  Inicpila(jugador1, '');
  Inicpila(jugador2, '');
  while not pilavacia (pozo) do
    begin
      apilar(jugador1, desapilar(pozo));
      if not pilavacia (pozo) then
        apilar(jugador2, desapilar(pozo));
    end;
  Writepila(pozo);
  Writepila(jugador1);
  Writepila(jugador2);

End.

/*****/
```

```
Program e9p1;  
  
Uses estructu;  
  
Var  
  a,b,aux,verdadero,falso:pila;  
  
Begin  
  Readpila(a);  
  Readpila(b);  
  Inicpila(aux,'1');  
  Inicpila(verdadero,'');  
  Inicpila(falso,'');  
  while not pilavacia (a) and not pilavacia (b) do  
    begin  
      apilar(aux,desapilar(a));  
      apilar(aux,desapilar(b));  
    end;  
  if pilavacia (a) and pilavacia (b) then  
    apilar(verdadero,desapilar(aux))  
  else  
    apilar(falso,desapilar(aux));  
  Writepila(verdadero);  
  Writepila(falso);  
  
End.  
  
/*****/
```

```
Program e10p1;

Uses estructu;

Var
  a,b,aux,verdadero,falso:pila;

Begin
  Readpila(a);
  Readpila(b);
  Inicpila(aux,'1');
  Inicpila(verdadero,'');
  Inicpila(falso,'');
  while not pilavacia (a) and not pilavacia (b) do
    begin
      if tope(a) = tope(b) then
        apilar(aux,desapilar(a));
        apilar(aux,desapilar(b));
      end;
      if pilavacia (a) and pilavacia (b) then
        apilar(verdadero,desapilar(aux))
      else
        apilar(falso,desapilar(aux));
      Writepila(verdadero);
      Writepila(falso);

End.
/*****/

Program e11p1;

Uses estructu;

Var
  modelo,dada,eliminar,aux:pila;

Begin
  Readpila(dada);
  Inicpila(modelo,'4 5 2');
  while not pilavacia (dada) do
    if tope(modelo) = tope(dada) then
      apilar(eliminar,desapilar(dada))
    else
      apilar(aux,desapilar(dada));
    while not pilavacia (aux) do
      apilar(dada,desapilar(aux));
  Writepila(modelo);
  Writepila(dada);
  writepila (eliminar);

End.
/*****/
```

```
Program e12p1;
{programa que saca de la pila dada, todos los elementos existentes en la
pila dada}
uses estructu;
var modelo, dada, auxmodelo, auxdada, basura:pila;
begin
  readpila(modelo);
  readpila(dada);
  inicpila(auxmodelo, '');
  inicpila(auxdada, '');
  inicpila(basura, '');
  while (not pilavacia (modelo)) do
    begin
      while (not pilavacia (dada)) do
        if (tope (dada) = (tope (modelo))) then
          apilar(basura, desapilar(dada))
        else
          apilar(auxdada, desapilar(dada));
          while (not pilavacia (auxdada)) do
            apilar(dada, desapilar(auxdada));
          apilar(auxmodelo, desapilar(modelo));
        end;
      while(not pilavacia (auxmodelo)) do
        apilar(modelo, desapilar(auxmodelo));
        writepila(modelo);
        writepila(dada);
      end.

```

```
Program e13p1;
{pasa los elementos mayores al tope de la pila limite, a mayores, y los
menores a menores}
uses estructu;
var limite, dada, mayores, menores:pila;
begin
  readpila(dada);
  inicpila(limite, '3 4 5');
  inicpila(mayores, '');
  inicpila(menores, '');
  while not pilavacia (dada) do
    if tope (dada) < (tope (limite)) then
      apilar(menores, desapilar (dada))
    else
      apilar(mayores, desapilar (dada));
    writepila(menores);
    writepila(mayores);
  end.

```

```
/*-----*/
```

```
Program e14p1;  
{determina si la cantidad de elementos de la pila dada, es par o impar}  
Uses estructu;  
  
Var  
  dada, aux, par, impar, auxdada1, auxdada2:pila;  
  
Begin  
  Readpila(dada);  
  Inicpila(aux, '1');  
  Inicpila(par, '');  
  Inicpila(impar, '');  
  Inicpila(auxdada1, '');  
  Inicpila(auxdada2, '');  
  while not pilavacia (dada) do  
    begin  
      apilar(auxdada1, desapilar(dada));  
      if not pilavacia (dada) then  
        apilar(auxdada2, desapilar(dada));  
      end;  
  while not pilavacia (auxdada1) and not pilavacia (auxdada2) do  
    begin  
      apilar(dada, desapilar(auxdada1));  
      apilar(dada, desapilar(auxdada2));  
    end;  
  if pilavacia (auxdada1) and pilavacia (auxdada2) then  
    apilar(par, desapilar(aux))  
  else  
    apilar(impar, desapilar(aux));  
  Writepila(par);  
  Writepila(impar);  
  
End.
```