

Υλοποίηση ουράς με χρήση πίνακα

```
program QueueExample;

const
  MaxSize = 10;

type
  Queue = record
    Items: array [1..MaxSize] of Integer;
    Front, Rear: Integer;
  end;

procedure InitializeQueue(var Q: Queue);
begin
  Q.Front := 0;
  Q.Rear := 0;
end;

function IsEmptyQueue(const Q: Queue): Boolean;
begin
  IsEmptyQueue := Q.Front = Q.Rear;
end;

function IsFullQueue(const Q: Queue): Boolean;
begin
  IsFullQueue := (Q.Rear + 1) mod MaxSize = Q.Front;
end;

procedure Enqueue(var Q: Queue; Element: Integer);
begin
  if not IsFullQueue(Q) then
  begin
    Q.Rear := (Q.Rear + 1) mod MaxSize;
    Q.Items[Q.Rear] := Element;
  end
  else
    writeln('Η ουρά είναι γεμάτη.');
```

```
end;

function Dequeue(var Q: Queue): Integer;
begin
  if not IsEmptyQueue(Q) then
  begin
    Q.Front := (Q.Front + 1) mod MaxSize;
    Dequeue := Q.Items[Q.Front];
  end
```

```
else
begin
  writeln('Η ουρά είναι άδεια.');
```

Dequeue := -1; // Τιμή που υποδηλώνει ότι η ουρά είναι άδεια

```
end;
end;
```

var

```
MyQueue: Queue;
i, element: Integer;
```

begin

```
InitializeQueue(MyQueue);
```

writeln('Προσθέστε στοιχεία στην ουρά:');

```
for i := 1 to 5 do
begin
  write('Στοιχείο ', i, ': ');
  readln(element);
  Enqueue(MyQueue, element);
end;
```

writeln('Αφαιρέστε στοιχεία από την ουρά:');

```
while not IsEmptyQueue(MyQueue) do
  writeln('Αφαιρείται: ', Dequeue(MyQueue));
```

readln;

```
end.
```