

13. Сортировка выбором

```

procedure SortByChoice(a: array of integer);
begin
  for var i := 0 to a.Length - 2 do
  begin
    var min := a[ i ];
    var imin := i;
    for var j := i + 1 to a.Length - 1 do
      if a[j] < min then
        begin
          min := a[j];
          imin := j;
        end;
    a[imin] := a[i];
    a[i] := min;
  end;
end;

```

14. Пузырьковая сортировка

```

procedure BubbleSort(a: array of integer);
begin
  for var i := 0 to a.Length - 2 do
    for var j := a.Length - 1 downto i + 1 do
      if a[j] < a[j - 1] then
        Swap(a[j], a[j-1]);
  end;

```

**procedure** BubbleSort2(a: **array of** integer);

```

begin
  var i := a.Length - 1;
  var q: boolean;
  repeat
    q := true;
    for var j := 0 to i - 1 do
      if a[j + 1] < a[j] then
        begin
          Swap(a[j + 1], a[j]);
          q := false;
        end;
    i -= 1;
  until q;
end;

```

15. Сортировка вставками

```

procedure SortByInsert(a: array of integer);
begin
  for var i:=1 to a.Length - 1 do
  begin
    var x := a[i];
    var j := i - 1;
    while (j >= 0) and (x < a[j]) do
      begin
        a[j + 1] := a[j];
        j -= 1;
      end;
    a[j + 1] := x;
  end;
end;

```

16. Поиск по условию

```

type
  IPredicate = function(x: integer): boolean;

function Even(x: integer): boolean;
begin
  Result := not odd(x);
end;

function IsPositive(x: integer): boolean;
begin
  Result := x > 0;
end;

function FindPred(a: array of integer;
  pred: IPredicate): integer;
begin
  var n := a.Length;
  var i := 0;
  while (i<n) and not pred(a[i]) do
    i += 1;
  if i = n then
    Result := -1
  else Result := i;
end;

```

17. Количество по условию

```

function CountPred(a: array of integer;
  pred: IPredicate): integer;
begin
  Result := 0;
  for var i := 0 to a.Length - 1 do
    if pred(a[i]) then
      Result += 1;
end;

```

18. Условный минимум

```

procedure MinElemPred(a: array of integer;
  pred: IPredicate; var min, imin: integer);
begin
  min := Integer.MaxValue;
  imin := -1;
  for var i:=0 to a.Length - 1 do
    if pred(a[i]) and (a[i] < min) then
      begin
        min := a[i];
        imin := i;
      end;
end;

```

19. Удаление по условию

```

procedure DeleteAll(a: array of integer;
  var n: integer; pred: IPredicate);
begin
  Assert((0<n) and (n<=a.Length));
  var j := 0;
  for var i := 0 to n - 1 do
    if not pred(a[i]) then
      begin
        a[j] := a[i];
        j += 1;
      end;
    n := j;
end;

```