**Поиск**

***program*** *Prim5\_3;*

***type***

*Index=1..4096;*

*Vector=****array****[Index]* ***of*** *Integer;*

***var***

*N,Svar : Integer;*

*I,L,R : Index;*

*Name :* ***string****;*

*W : Vector;*

*F :* ***file of*** *Integer;*

***begin***

*ClrScr;*

*Write ('Введите путь и имя файла ');*

*Read (Name);*

*Write ('Введите значение искомой компоненты ');*

*Read (SVar);*

*Assign (F, ‘A:\+Name’);*

*Reset (F);*

*N :=FileSize(F);*

***if*** *N > 4096*

***then***

*Write ('Ошибка в исходных данных')*

***else***

***begin***

***for*** *I :=1* ***to*** *N* ***do*** *{цикл формирования вектора}*

*Read(F,W[I]);*

*L :=1;*

*R :=N;*

***while (****W[I]<>SVar)* ***and*** *(L<=R)* ***do***

***begin***

*I :=(L+R)* ***div*** *2;*

***if*** *SVar < W[I]*

***then***

*R :=I - 1*

***else***

*L :=I+1*

***end;***

***if*** *W[I]=SVar*

***then***

*WriteLn('Искомое значение находится в ',i,'-й компоненте')*

***else***

*WriteLn('В векторе нет искомой компоненты')*

**end;**

**close** *(F);*

*ReadKey*

***end****. {Prim5\_3}*

**Сортировка простыми включениями**

***procedure*** *Straightinsertion;*

***var***

*I,J : Index;*

*X : Item;*

***begin***

***for*** *I :=2* ***to*** *N* ***do***

***begin***

*X :=W[I];*

*W[0] :=X;*

*J :=I - 1;*

***while*** *X.Key < A[J].Key* ***do***

***begin***

*W[J+1] :=W[J];*

*J :=J - 1;*

***end;***

*W[J+1] :=X*

***end***

***end****; {Straightinsertion}*

**Сортировка бинарными включениями.**

***procedure*** *Binaryinsertion;*

***var***

*I,J,L,R,M : Index;*

*X : Item;*

***begin***

***for*** *I :=2* ***to*** *N* ***do***

***begin***

*X :=W[I];*

*L :=1;*

*R :=I - 1;*

***while*** *L<=R* ***do***

***begin***

*M:=(L+R)* ***div*** *2;*

***if*** *X.Key<W[M].Key*

***then***

*R :=M - 1*

***else***

*L :=M+1*

***end;***

***for*** *J :=I - 1* ***downto*** *L* ***do***

*W[J+1] :=W[J];*

*W[L] :=X*

***end***

***end****; {Binaryinsertion}*

**Сортировка простым обменом**

**Procedure** *Bubblesort***;**

**var**

*I,J : Index;*

*X : Item;*

**begin**

**for** *I :=2* **to** *N* **do**

**begin**

**for** *J :=N* **downto** *I* **do**

**if** *A[J - 1].Key > A[J].Key* **then**

**begin**

*X :=A[J-1];*

*A[J-1] :=A[J];*

*A[J] :=X;*

**end;**

**end;**

**end;**

**Шейкер-сортировка**

**Procedure** *Shakesort;*

**var**

*J,K,L,R : Index;*

*X : Item;*

**begin**

*L :=2;*

*R :=N;*

*K :=N;*

**repeat**

**for** *J :=R* **downto** *L* **do**

**if** *A[J - 1].Key > A[J].Key* **then**

**begin**

*X :=A[J - 1];*

*A[J - 1] :=A[J];*

*A[J] :=X;*

*K :=J;*

**end;**

*L :=K+1;*

**for** *J :=L* **to** *R* **do**

**if** *A[J - 1].Key > A[J].Key* **then**

**begin**

*X :=A[J - 1];*

*A[J - 1] :=A[J];*

*A[J] :=X;*

*K :=J;*

**end;**

*R :=K - 1;*

**until** *L > R;*

**end;**

**Сортировка простым выбором.**

**procedure** *Straightselection***;**

**var**

*I,J,K : Index;*

*X : Item;*

**begin**

**for** *I :=1* **to** *N -1* **do**

**begin**

*K :=I;*

*X :=W[K];*

***for*** *J :=I+1* ***to*** *N* ***do***

***if*** *W[J].Key <X.Key* ***then***

***begin***

*K :=J;*

*X :=W[J]*

***end****;*

*W[K] :=W[I];*

*W[I] :=X*

***end***

***end****; {Straightselection}*

**Сортировка включениями с убывающим приращением**

**procedure** *Shellsort***;**

**const**

*T = 4;*

**var**

*I,J,K,S : Index;*

*M : 1 .. T;*

*X : Item;*

*H :* **array***[1 .. T]* **of** *Integer***;**

**begin**

*H[1] := 9;*

*H[2] := 5;*

*H[3] := 3;*

*H[4] := 1;*

**for** *M := 1* **to** *T* **do**

**begin**

*K := H[M];*

*S := -K; {место барьера}*

**for** *I := K+1* **to** *N* **do**

**begin**

*X := W[I];*

*J := I - K;*

**if** *S = 0* **then**

*S := -K;*

*S := S+1;*

*W[S] := X;*

**while** *X.Key < W[J].Key* **do**

**begin**

*W[J+K] := W[J];*

*J := J - K*

**end;**

*W[J+K] := X*

**end**

**end**

**end;**

**Сортировка с помощью дерева**

***procedure*** *Heapsort;*

***var*** *L,R: Index;*

*X : Item;*

***procedure*** *Sift;*

***var*** *I,J : Index;*

***begin***

*I := L;*

*J := 2 \* I;*

*X := A[I];*

***while*** *J <= R* ***do***

***begin***

***if*** *J < R* ***then***

***if*** *A[J].Key < A[J+1].Key* ***then***

*J := J+1;*

***if*** *X.Key >= A[J].Key* ***then*** *Break;*

*A[I] := A[J];*

*I := J;*

*J := 2 \* I*

***end*** *;*

*A[I] := X*

***end*** *;*

***begin***

*L := (N* ***div*** *2) + 1;*

*R := N;*

***while*** *L > 1* ***do***

***begin***

*L := L - 1;*

*Sift*

***end*** *;*

***while*** *R > 1* ***do***

***begin***

*X := A[L];*

*A[L] := A[R];*

*A[R] := X;*

*R := R - 1;*

*Sift*

***end***

***end;*** *{Heapsort}*

**Сортировка с разделением.**

***procedure*** *Quicksort;*

***procedure*** *Sort (L,R : Index);*

***var*** *I,J : Index;*

*X,W : Item;*

***begin***

*I := L; J := R;*

*X := A[(L+R)* ***div*** *2];*

***repeat***

***while*** *A[I].Key < X.Key* ***do*** *I := I+1;*

***while*** *X.Key < A[J].Key* ***do*** *J := J - 1;*

***if*** *I <= J* ***then***

***begin***

*W := A[I];*

*A[I] := A[J];*

*A[J] := W;*

*I := I+1;*

*J := J - 1*

***end***

***until*** *I > J;*

***if*** *L < J* ***then*** *Sort(L,J);*

***if*** *I < R* ***then*** *Sort(I,R)*

***end*** *;*

***begin***

*Sort(1,N)*

***end;*** *{Quicksort}*

**Поиск медианы**

***procedure*** *Find (K :**Integer);*

***var***

*L,R,I,J,W,X : Integer;*

***begin***

*L := 1;*

*R := N;*

***while*** *L < R* ***do***

***begin***

*X := A[K];*

*I := L;*

*J := R;*

***repeat*** *{split}*

***while*** *A[I] < X* ***do***

*I := I+1;*

***while*** *X < A[J]* ***do***

*J := J-1;*

***if*** *I < J* ***then***

***begin***

*W := A[I];*

*A[I] := A[J];*

*A[J] := W;*

*I := I+1;*

*J := J - 1*

***end***

***until*** *I > J;*

***if*** *J < K* ***then*** *L := I;*

***if*** *K < I* ***then*** *R := J*

***end***

***end****; {Find}*