program array\_processing;

uses crt;

const limit =5; rogue\_value = -1;

type array\_type = array [1..limit] of integer;

var table :array\_type;count:integer;

procedure initialise\_array;

var i:integer;

begin

clrscr;

for i:= 1 to limit do

table[i]:= 0

end;

Procedure load\_array;

var num:integer;

begin

gotoxy(10,10);

writeln('Inputting data into the Array');

writeln;

count:=0;

write('Please enter number -> '); readln(num);

while (num <> rogue\_value) and ( count <> limit) do

begin

count:=count + 1;

table[count]:=num;

write('Please enter number -> '); readln(num)

end;

end;

Procedure print\_array;

var i:integer;

begin

clrscr;

gotoxy(10,10);

writeln(' Contents of Array');

writeln;

for i:= 1 to limit do

writeln(' ',table[i]);

readln;

end;

procedure process\_array;

var average:real; largest,sum,i:integer;

begin

sum:=0; largest:= table[1];

for i:= 1 to limit do

begin

sum:= sum+ table[i];

if (table[i] > largest) then

largest:= table[i];

end;

average:= sum/ limit;

clrscr;

gotoxy(20,10);

writeln('Average of contents of the array : ',average:4:2);

gotoxy(20,11);

writeln('The largest number in the array : ',largest);

readln;

end;

procedure sequential\_search;

var i,num,location:integer;

found:boolean;

begin

clrscr;

gotoxy(20,10);

write('please enter number to locate '); readln(num);

found := false;

for i:= 1 to count do

begin

if (table[i] = num) then

begin

found:= true;

location:=i;

i:= count;

end;

end;

gotoxy(20,11);

if (found = false) then

writeln(num,' does not exist')

else

writeln(num,' found in location ',location);

readln;

end;

procedure binary\_search;

var low,high,middle,num:integer;

begin

clrscr;

gotoxy(20,10);

writeln('Enter num to locate ');readln(num);

low:=1; high:=count;

repeat

middle:= (low + high) div 2;

if (num > table[middle]) then

low := middle + 1

else

if (num < table[middle]) then

high:= middle -1

until (num = table[middle]) or (low > high);

if (low > high) then

writeln(num,' does not exist')

else

writeln(num,' found in location ',middle);

readln;

end;

begin {.................main program ....................}

initialise\_array;

load\_array;

print\_array;

process\_array;

sequential\_search;

(\* binary\_search; can only be used if the data is sorted\*)

end. {.................main program .....................}