



Province of the  
**EASTERN CAPE**  
EDUCATION

**NATIONAL  
SENIOR CERTIFICATE**

**GRADE 11**

**NOVEMBER 2017**

**INFORMATION TECHNOLOGY P1  
MARKING GUIDELINE**

**MARKS: 150**

---

This marking guideline consists of 14 pages.

---

**GENERAL PROGRAMMING SKILLS**

QUESTION 1	MAX. MARKS	MARKS ACHIEVED
<p>Get input from spinedits for adults, ✓ children✓ and pensioners✓</p> <p>If VIP checkbox is selected✓ get number of VIP tickets from spinedit✓</p> <p><u>Allocation of ticket numbers: (8)</u></p> <p>Initialising of variables✓ VIP tickets start at 1✓ Other tickets start at 31✓ Correct allocation of ticket numbers – start ✓ and end✓ Calculation of total VIP tickets ✓ Calculation of total other tickets✓ Calculation of total tickets for all purchases made ✓</p> <p><u>Calculation of the cost: (4)</u></p> <p>Ticket costs per category – adults * 80✓ and (children+pensioners * 50) ✓ Calculating total cost per purchase✓ Calculating total cost for all purchases made✓</p> <p><u>Display: (5)</u></p> <p>Heading: SCHOOL PRODUCTION✓ and on next line Show:GALA NIGHT✓ Cost of tickets✓ formatted to currency and two decimal places✓ The ticket numbers that are allocated to this purchase✓</p> <p><u>Write to File: (6)</u></p> <p>Assignfile✓ Rewrite✓ Write total amount of tickets to file✓ Write total amount of money to file,✓ formatted to two decimal places✓ Closefile✓</p> <p><u>Clear button: (4)</u></p> <p>Clear the richedit✓ Set the spinedits to 0✓✓ Set the checkbox to uncheck✓</p>	<b>32</b>	
	<b>[32]</b>	

QUESTION 2		MAXIMUM MARKS	MARKS ACHIEVED
2.1	Class scope variable declarations: icount✓ arrnames✓, arrsurnames✓, arrscene✓[1..100] ✓ string✓  Test if file exists✓ Message to display that the file does not exist✓ Close program✓ Assignfile ✓ Reset file✓ Initialise the counter✓ Loop ✓ Increase the counter✓ Read line from text file✓ Find position of space✓ Get the name✓ and store in array✓ Find position of ' '✓ Get the surname and store in array✓ Store the scene in array✓ Close the file✓	22	
2.2	Method heading✓ Outer loop (icount – 1) ✓ Inner loop (k+1) ✓ Compare <b>arrsurnames</b> ✓ 3 lines to swop arrnames✓✓✓ 3 lines to swop arrsurnames✓ 3 lines to swop arrscene✓  * the first array that is swopped will be awarded 3 marks because the next two make use of the same logic – must make use of a temporary variable	9	
2.3	Call the Sort method✓ Loop – making use of counter variable✓ Displaying the surnames in uppercase✓ and the names✓	4	
2.4	Initialise counters✓ Loop✓ Any method to determine if S1 is present✓ Increase counter✓ Any method to determine if S2 is present✓ Increase counter✓ Display how many people in S1✓ and in S2✓	8	
2.5	Call the Sort method✓ Loop✓ If Technical✓ or Backstage✓ Display arrname ' ' arrsurname ' '✓ arrscene✓	6	
		<b>[49]</b>	

QUESTION 3		MAX MARKS	MARKS ACHIEVED
3.1	<p><b>ValidId</b></p> <p>Method heading: id as parameter✓ and Boolean return type✓</p> <p>Initialise counters✓</p> <p>Get the numbers in odd positions✓ and add them✓</p> <p>Get the numbers in even positions✓, multiply by 2✓ and put numbers together✓</p> <p>Add all the numbers (even) ✓</p> <p>Add odd and even✓</p> <p>Get the last digit✓</p> <p>Subtract last digit from 10✓</p> <p>If✓ answer is the same as the last digit in ID✓</p> <p>    Return true✓</p> <p>else✓</p> <p>    Return false✓</p>	17	
3.2	<p><b>Gender</b></p> <p>Method heading: Id as parameter✓ return type char✓</p> <p>If 7<sup>th</sup> position of ID✓ less than or equal to 4✓</p> <p>    Return 'F'✓</p> <p>else✓</p> <p>    Return 'M'✓</p>	7	
3.3	<p><b>RegCode</b></p> <p>Procedure✓ with sname as parameter✓</p> <p>Initialise variables: icount := 0✓; k := 1✓; scode := "" ✓</p> <p>Conditional loop✓</p> <p>    Change name to uppercase✓</p> <p>    If letter not a vowel✓ or space✓</p> <p>        Add to code✓</p> <p>        Increase counter (icount) ✓</p> <p>        Increase counter (k) ✓</p> <p>Randomise a number✓ correct range ✓ (100-999)</p> <p>Display code✓, #✓ and random number✓</p>	17	
3.4	<p>Get the id✓</p> <p>Get the name✓</p> <p>Get the gender✓</p> <p>If the ID is valid✓ and gender from ID = gender from input✓</p> <p>    Call Regcode✓</p> <p>else✓</p> <p>    Display a message that information is not correct✓</p>	8	
		<b>[49]</b>	

QUESTION 4		MAX MARKS	MARKS ACHIEVED
4.1	Sort according surname ASC✓, name ASC✓	2	
4.2	NO FILTERS!  Get the name and surname from inputbox✓ Extract the name✓ Extract the surname✓ Open the table and set to first record✓ Loop through table✓ If name ✓ and surname✓ is found Display the contact number✓ Go to next record✓	9	
4.3	Any method may be used. If a filter was used, the filtered property should be set to false in the next question.  Using the word LIKE✓ %Cast Welfare%✓	2	
4.4	Any method may be used Using recordcount or a counter✓ Displaying how many records✓	2	
4.5	Insert✓ Add the name✓ Add the surname✓ Add the contact number✓ Post✓	5	
		<b>[20]</b>	
<b>TOTAL:</b>		<b>150</b>	

**SAMPLE SOLUTIONS****Question 1**

```
unit Question1_u;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, StdCtrls, ComCtrls, ExtCtrls, Spin, Math, Buttons;
```

```
type
```

```
TForm1 = class(TForm)  
    sedadults: TSpinEdit;  
    sedchildren: TSpinEdit;  
    redoutput: TRichEdit;  
    lbladults: TLabel;  
    Label1: TLabel;  
    Button1: TButton;  
    cbxVip: TCheckBox;  
    Label2: TLabel;  
    sedpens: TSpinEdit;  
    BitBtn1: TBitBtn;  
    procedure Button1Click(Sender: TObject);  
    procedure FormCreate(Sender: TObject);  
    procedure BitBtn1Click(Sender: TObject);
```

```
private
```

```
    { Private declarations }
```

```
public
```

```
    { Public declarations }
```

```
end;
```

```
var
```

```
Form1: TForm1;  
itotalvip, itotalother, itotaltickets : integer;  
rtotalcost : real;
```

```
implementation
```

```
{ $R *.dfm }
```

```
procedure TForm1.BitBtn1Click(Sender: TObject);
```

```
begin
```

```
    redoutput.clear;  
    sedadults.Value := 0;  
    sedchildren.Value := 0;  
    sedpens.Value := 0;  
    cbxVip.Checked := false;
```

```
end;
```

```
procedure TForm1.Button1Click(Sender: TObject);
var
  iadults, ildren, ipens, ivip, istartseat, iendseat : integer;
  rcost : real;
  myfile : textfile;

begin
  ivip := 0;
  iadults := sedadults.Value;
  ipens := sedpens.value;
  ildren := sedchildren.Value;

  istartseat := itotalother;
  iendseat := istartseat + iadults + ipens + ildren -1;

  if cbxvip.Checked = true then
    begin
      ivip := strtoint(Inputbox(", 'How many VIPs?',"));
      istartseat := itotalvip;
      iendseat := istartseat + ivip-1;
    end;
  itotalvip := itotalvip + ivip;
  itotalother := itotalother + iadults + ipens + ildren;

  rcost := iadults * 80 + (ipens + ildren) * 50;
  rtotalcost := rtotalcost + rcost;
  itotaltickets := itotaltickets + ivip + iadults + ildren + ipens;

  redoutput.Lines.Add('School Production');
  redoutput.Lines.Add('Show: GALA NIGHT');
  redoutput.lines.add('Total cost of tickets: '+#9+floattostrf(rcost,ffcurrency,10,2));
  redoutput.Lines.Add('Seats: '+inttostr(istartseat) + '-' +inttostr(iendseat));

  Assignfile(myfile, 'Bookings.txt');
  Rewrite(myfile);
  writeln(myfile, 'Tickets sold: '+inttostr(itotaltickets));
  writeln(myfile, 'Total amount '+floattostrf(rtotalcost,ffixed,6,2));
  closefile(myfile);
  showmessage('File was successfully written');

end;

procedure TForm1.FormCreate(Sender: TObject);
begin
  itotalvip := 1;
  itotalother := 31;
  rtotalcost := 0;
  itotaltickets := 0;
end;

end.
```

**Question 2**

```
unit Question2_u;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs,  
StdCtrls;
```

```
type
```

```
TForm1 = class(TForm)  
  Button1: TButton;  
  Memo1: TMemo;  
  Button2: TButton;  
  Button3: TButton;  
  procedure FormCreate(Sender: TObject);  
  procedure Button1Click(Sender: TObject);  
  procedure Button2Click(Sender: TObject);  
  procedure Button3Click(Sender: TObject);  
private  
  { Private declarations }  
public  
  procedure Sort;  
end;
```

```
var
```

```
Form1: TForm1;  
arrnames, arrsurnames, arrscene : array[1..100] of string;  
icount : integer;
```

```
implementation
```

```
{ $R *.dfm }
```

```
procedure TForm1.Button1Click(Sender: TObject);
```

```
var
```

```
  k : integer;
```

```
begin
```

```
  Sort;
```

```
  for k := 1 to icount do
```

```
    memo1.Lines.Add(uppercase (arrsurnames[k]) + ' ' + arrnames[k]);
```

```
end;
```

```
procedure TForm1.Sort;
```

```
var
```

```
  k: Integer;
```

```
  l: Integer;
```

```
  stemp1, stemp2, stemp3 : string;
```

```
begin
```

```
  for k := 1 to icount - 1 do
```

```
    for l := k+1 to icount do
```

```
      if arrsurnames[k] > arrsurnames[l] then
```

```
        begin
```



```
    stemp1 := arrnames[k];
    arrnames[k] := arrnames[l];
    arrnames[l] := stemp1;
    stemp2 := arrsurnames[k];
    arrsurnames[k] := arrsurnames[l];
    arrsurnames[l] := stemp2;
    stemp3 := arrscene[k];
    arrscene[k] := arrscene[l];
    arrscene[l] := stemp3;
end;
end;
procedure TForm1.Button2Click(Sender: TObject);
var
    k, ipos, iscene1, iscene2 : integer;
begin
    memo1.Clear;
    iscene1 := 0;
    iscene2 := 0;
    for k := 1 to icount do
    begin
        ipos := pos('S1',arrscene[k]);
        if ipos > 0 then
            inc(iscene1);

        ipos := pos('S2',arrscene[k]);
        if ipos > 0 then
            inc(iscene2);

    end;
    memo1.Lines.Add('Total Scene 1: ' + inttostr(iscene1));
    memo1.Lines.Add('Total Scene 2: ' + inttostr(iscene2));
end;

procedure TForm1.Button3Click(Sender: TObject);
var
    k : integer;
begin
    memo1.Clear;
    Sort;
    for k := 1 to icount do
    begin
        if (arrscene[k] = 'Technical') or (arrscene[k] = 'Backstage') then
            memo1.Lines.Add(arrnames[k] + ' '+arrsurnames[k] + '-' + arrscene[k])
        end;
    end;
end;

procedure TForm1.FormCreate(Sender: TObject);
var
    myfile : textfile;
    soneline : string;
    ipos : integer;
begin
    if fileexists('Cast.txt') <> true then
```

```
begin
  Showmessage('File does not exist');
  Exit;
end;
Assignfile(myfile, 'Cast.txt');
Reset(myfile);
icount := 0;
while not eof (myfile) do
begin
  icount := icount + 1;
  readln(myfile,soneline);
  ipos := pos(' ',soneline);
  arrnames[icount] := copy(soneline, 1, ipos-1);
  delete(soneline,1,ipos);
  ipos := pos('-',soneline);
  arrsurnames[icount] := copy(soneline, 1, ipos-1);
  delete(soneline,1,ipos);
  arrscene[icount] := soneline;
end;
closefile(myfile);
end;

end.
```

**Question 3**

```
unit Question3_u;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, StdCtrls, Math;
```

```
type
```

```
TForm1 = class(TForm)  
    Button1: TButton;  
    Edit1: TEdit;  
    Edit2: TEdit;  
    Edit3: TEdit;  
    procedure Button1Click(Sender: TObject);  
private  
    { Private declarations }  
public  
    function validid(sid : string) : boolean;  
    function gender(sid : string) : char;  
    Procedure RegCode(sname : string);  
end;
```

```
var
```

```
Form1: TForm1;
```

```
implementation
```

```
{$R *.dfm}
```

```
procedure TForm1.Button1Click(Sender: TObject);
```

```
var
```

```
sid, sname, sgender : string;
```

```
begin
```

```
sid := edit1.text;
```

```
sname := edit2.text;
```

```
sgender := edit3.text;
```

```
if (validid(sid)) and (gender(sid)= sgender) then
```

```
    regcode(sname)
```

```
else
```

```
    Showmessage('Information is not correct');
```

```
end;
```

```
procedure TForm1.RegCode (sname: string);
```

```
var
```

```
icount, k, iran : integer;
```

```
scode : string;
```

```
begin
```

```
icount := 0;
```

```
k := 1;
```

```
scode := '';
```

```
while icount <> 4 do
begin
  if NOT(upcase (sname[k]) in ['A','E','I','O','U',' ']) then
  begin
    scode := scode + sname[k];
    inc(icount);
  end;
  inc(k);
end;
iran := randomrange(1,1000);
Showmessage(scode+'#+inttostr(iran) );
end;
```

```
function TForm1.gender(sid: string): char;
begin
  if strtoint(sid[7] ) <= 4 then
    result := 'F'
  else
    result := 'M';
end;
```

```
function TForm1.validid(sid: string): boolean;
var
  k, iodd, ieven, isum, icode, ilast: Integer;
  seven : string;
begin
  seven := "";
  ieven := 0; initialise counters
  iodd := 0;
  for k := 1 to 12 do
  begin
    if k mod 2 = 0 then
      seven := seven + inttostr(strtoint(sid[k])*2)
    else
      iodd := iodd + strtoint(sid[k]);
    end;
  for k := 1 to length(seven) do
    ieven := ieven + strtoint(seven[k]);
  end;

  isum := iodd + ieven;
  ilast := strtoint(inttostr(isum)[2]);

  icode := 10 - ilast;

  if icode = strtoint(sid[13]) then
    result := true
  else
    result := false;
end;

end.
```

**Question 4**

```
unit Question4_u;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, Menus, Grids, DBGrids, DB, ADODB;
```

```
type
```

```
TForm1 = class(TForm)  
  ADOTable1: TADOTable;  
  DataSource1: TDataSource;  
  DBGrid1: TDBGrid;  
  MainMenu1: TMainMenu;  
  Sort1: TMenuItem;  
  FindaContactNumber1: TMenuItem;  
  CastWelfare1: TMenuItem;  
  AllHelpers1: TMenuItem;  
  Remove1: TMenuItem;  
  procedure Sort1Click(Sender: TObject);  
  procedure FindaContactNumber1Click(Sender: TObject);  
  procedure CastWelfare1Click(Sender: TObject);  
  procedure AllHelpers1Click(Sender: TObject);  
  procedure Add1Click(Sender: TObject);  
private  
  { Private declarations }  
public  
  { Public declarations }  
end;
```

```
var
```

```
Form1: TForm1;
```

```
implementation
```

```
{$R *.dfm}
```

```
procedure TForm1.AllHelpers1Click(Sender: TObject);  
begin  
  Showmessage(inttostr(adotable1.RecordCount));  
end;
```

```
procedure TForm1.CastWelfare1Click(Sender: TObject);  
begin  
  adotable1.Filter := 'Tasks LIKE "%'+Cast Welfare+'%";  
  ADOTable1.Filtered := True;  
end;
```

```
procedure TForm1.FindaContactNumber1Click(Sender: TObject);  
var  
  sname, ssurname, ssearch : string;
```

```
ipos : integer;
begin
  ssearch := inputbox(",","");
  ipos := pos(' ',ssearch);
  sname := copy(ssearch, 1, ipos-1);
  delete(ssearch,1,ipos);
  ssurname := ssearch;
  adotable1.Open;
  adotable1.first;
  while not adotable1.eof do
    begin
      if (adotable1['Name'] = sname) and (adotable1['Surname'] = ssurname) then
        Showmessage('Contact Number is '+adotable1['ContactNumber']);
      adotable1.Next;
    end;
  adotable1.Close;
end;

procedure TForm1.Add1Click(Sender: TObject);
begin
  adotable1.Insert;
  adotable1['Name'] := 'Paige';
  adotable1['Surname'] := 'Watson';
  adotable1['ContactNumber'] := '0832454216';
  adotable1.Post;
end;

procedure TForm1.Sort1Click(Sender: TObject);
begin
  adotable1.Sort := 'Surname ASC, Name ASC';
end;

end.
```