

Form 6 ICT SBA – Case Study 3

Seating plan

Form 6 _____ Name _____ No. _____

Date _____

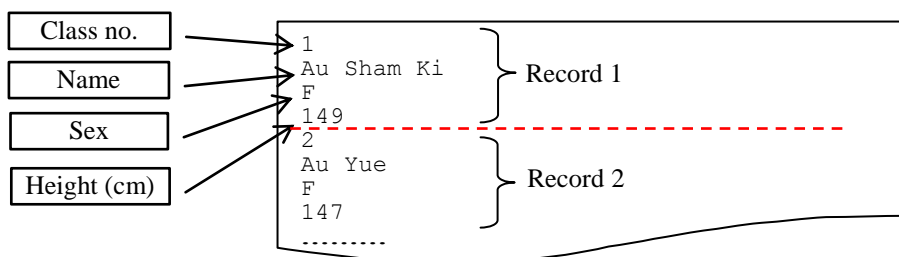
In ABC school, there are two types of classrooms, namely typical classrooms and special rooms. In a typical classroom, the seats are arranged in 5 rows and 8 columns. In a special room, there are 5 tables each can accommodate up to 8 students. Assume that each class has exactly 40 students.

The aim of this Case Study is to design a Pascal program to allocate the seats of a class of 40 students in a typical classroom and a special room.

The followings are the common ways of allocating seats:

- The seats are allocated in a completely random manner.
- The seats are allocated to boys and girls alternatively so that every neighbor of a boy is a girl or vice versa. You may assume that there approximately equal numbers of boys and girls in a class.
- The front seats are allocated to shorter students and the seats at the back are allocated to tall students.

The data of a class of students are stored in a text file with the following format:



You are required to write a Pascal program to achieve the following tasks:

1. Read the data of a class from a text file (e.g. class.txt).

The data are stored in the following parallel arrays in the program:

- classno : array[1..num_stud] of integer;
- name : array[1..num_stud] of string[250];
- sex : array[1..num_stud] of char;
- height : array[1..num_stud] of integer;

2. Display a menu for the user to choose different functions of the program

Seating Plan

=====

- (1) Sort by class number
- (2) Sort by height (ascending)
- (3) Sort randomly
- (4) Arrange boys and girls in alternate seats
- (5) Display name list
- (6) Display classroom seating plan
- (7) Display seating plan of special room (5 tables)
- (8) Read another class
- (9) Quit

Enter your choice:

3. Design different procedures to perform the tasks provided on the main menu. Sample outputs of choosing options (5), (6) and (7) are given below.

Option (5) (after choosing options (2) and (4))

No.	Name	sex	Height
34	Ngai Yat To	M	135
7	Chan Tai Man	F	137
3	Chan Kai Bong	M	136
6	Chan Shui Wah	F	138
.....			
<<< Press <Enter> to return. >>>			

Option (6)

Classroom Seating Plan							
=====							
22	15	35	39	9	11	32	21

28	23	40	18	37	30	20	38

33	25	19	29	26	1	4	10

8	27	5	14	17	16	31	2

34	7	3	6	36	24	13	12

Teacher's Desk							

<<< Press <Enter> to return. >>>							

Option (7)

Table 1 :			
=====			
No.	Name	sex	Height
=====			
34	Ngai Yat To	M	135
7	Chan Tai Man	F	137
3	Chan Kai Bong	M	136
6	Chan Shui Wah	F	138
36	Tsang King Fung	M	136
24	Kwok Foo Ho	F	141
13	Cheung Yee Hung	M	137
12	Cheung Koo Ho	F	142
=====			
Table 2 :			
.....			
<<< Press <Enter> to return. >>>			

Data Structures

The following data structures will be used in the program:

- For storing data of students in a class. The data of are from a text file (e.g. class.txt).
 - Parallel arrays:
 - classno : array[1..num_stud] of integer;
 - name : array[1..num_stud] of string[250];
 - sex : array[1..num_stud] of char;
 - height : array[1..num_stud] of integer;
- For storing the corresponding class numbers of the students allocated to the seats in a typical classroom.
 - A 2-dimensional array:
 - seat : array[1..5,1..8] of integer;

:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
seat[3,1]	:	:	:	:	:	:	:
seat[2,1]	:	:	:	:	:	:	:
seat[1,1]	seat[1,2]	seat[1,3]	:	:	:	:	:

Teacher's Desk

Procedures

The program consists of the following main procedures

- procedure read_data**
 - Read the data of a class from the text file (e.g. class.txt) and store them into the parallel arrays.
- procedure display_student_list;**
 - Display the current records in the parallel arrays according to the index of the arrays.
- procedure swap(i, j : integer)**
 - Swap the records with indices i and j in the parallel arrays.
- procedure sort_by_classno**
 - Sort the records in the parallel arrays in ascending order of the class number.

- **procedure sort_by_height**
 - Sort the records in the parallel arrays in ascending order of the height.
- **procedure sort_randomly**
 - Sort the records in the parallel arrays randomly.
- **procedure alternate_boy_girl**
 - Arrange the records in the parallel arrays with boys and girls in alternate c
- **procedure assign_classroom_seat**
 - Assign the class numbers (classno[.]) to the classroom seats (seat[. , .]).
- **procedure classroom_seating_plan;**
 - Display the classroom seating plan (seat[. , .]).
- **procedure special_room_seating_plan**
 - Display the special room seating plan.

Complete the program “seating_plan.pas” by referring to the results on running the executable program “seating_plan.exe”.

Complete the program below by referring to the results on running the executable program (seating_plan.exe).

```

program SeatingPlan;
uses Crt;
const
    num_stud = 40;
var
    classno : array[1..num_stud] of integer;
    name : array[1..num_stud] of string[25];
    sex : array[1..num_stud] of char;
    height : array[1..num_stud] of integer;
    seat : array[1..5,1..8] of integer;
    choice : char;

procedure read_data;
var
    filename : string;
    f : text;
    i : integer;
begin
    clrscr;
    write('Enter the filename of the class file: ');
    readln(filename);
    assign(f, filename);
    reset(f);
    for i := 1 to num_stud do
        begin
            readln(f, classno[i]);
            readln(f, name[i]);
            readln(f, sex[i]);
            readln(f, height[i]);
        end;
    close(f)
end;

procedure display_student_list;
var
    i : integer;
begin
    clrscr;
    writeln('No.':4, ' ', 'Name', '':21, 'Sex', ' ', 'Height');
    writeln('=====');
    for i := 1 to num_stud do
        writeln(classno[i]:3, ' ', name[i], '':25-length(name[i]), sex[i]:2, height[i]:8);
    writeln('=====');
    writeln;
    write('<<< Press <Enter> to return. >>>');
    readln
end;

```

```

{To swap the records i and j in the parallel arrays}
procedure swap(i, j : integer);
var
    tmp_classno : integer;
    tmp_name : string[25];
    tmp_sex : char;
    tmp_height : integer;
begin
    tmp_classno := classno[i];
    tmp_name := name[i];
    tmp_sex := sex[i];
    tmp_height := height[i];
    classno[i] := classno[j];
    name[i] := name[j];
    sex[i] := sex[j];
    height[i] := height[j];
    classno[j] := tmp_classno;
    name[j] := tmp_name;
    sex[j] := tmp_sex;
    height[j] := tmp_height;
end;

```

```

procedure sort_by_classno;
var
    i, j : integer;
begin

```

```

    display_student_list
end;

```

```

procedure sort_by_height;
var
    i, j : integer;
begin

```

```

    display_student_list
end;

```

```

{To shuffle 1000 pairs of randomly selected records in the arrays}
procedure sort_randomly;
var
    i, k, p : integer;
begin
    randomize;
    for i := 1 to 500 do
        begin
            k := random(num_stud)+1;
            p := random(num_stud)+1;
            swap(k,p)
        end;
    display_student_list
end;

```

```
procedure alternate_boy_girl;
```

```
procedure assign_classroom_seat;
```

```
procedure classroom_seating_plan;
var
  i, j : integer;
begin
  assign_classroom_seat;
  clrscr;
  writeln('          Classroom Seating Plan          ');
  writeln('          =====                          ');
  writeln(' ----- ');
```

```
  writeln('          -----                          ');
  writeln('          | Teacher''s Desk |                ');
  writeln('          -----                          ');
  writeln;
  write('<<< Press <Enter> to return. >>>');
  readln
end;
```

```
procedure special_room_seating_plan;
```

```

begin { Main program }
  read_data;
  repeat
    clrscr;
    writeln('Seating Plan');
    writeln('=====');
    writeln;
    writeln('*****');
    writeln('(1) Sort by class number');
    writeln('(2) Sort by height (ascending)');
    writeln('(3) Sort randomly');
    writeln('(4) Arrange boys and girls in alternate seats');
    writeln('(5) Display name list');
    writeln('(6) Display classroom seating plan');
    writeln('(7) Display seating plan of special room (5 tables)');
    writeln('(8) Read another class');
    writeln('(9) Quit');
    writeln('*****');
    writeln;
    write('Enter your choice: ');
    readln(choice);
    writeln;
    case choice of
      '1' : sort_by_classno;
      '2' : sort_by_height;
      '3' : sort_randomly;
      '4' : alternate_boy_girl;
      '5' : display_student_list;
      '6' : classroom_seating_plan;
      '7' : special_room_seating_plan;
      '8' : read_data;
    end;
  until choice = '9'
end.

```