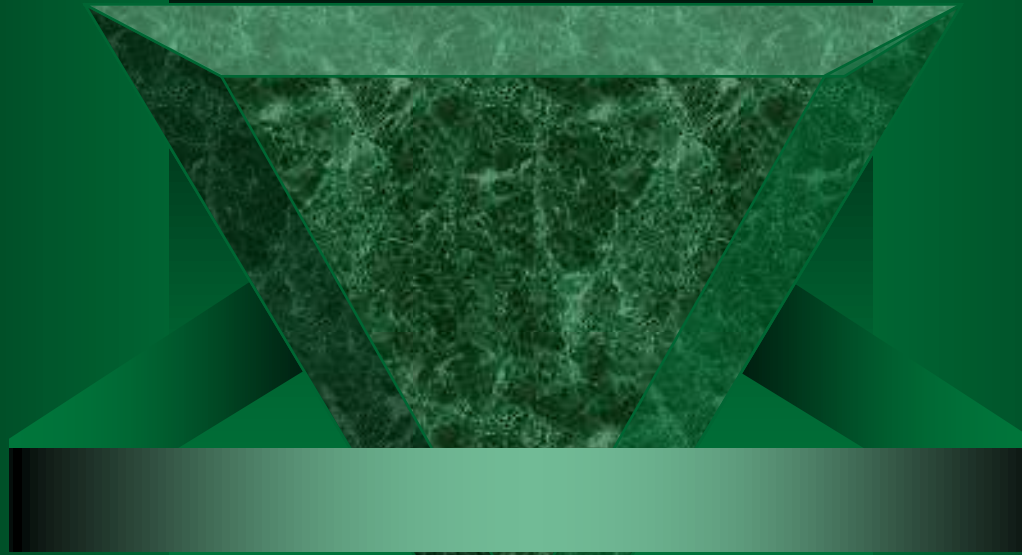


Apirak :



5. Structure



The Objectives

- Understand what structures are
- how to use the structures in Pascal

ตัวอย่าง 1

สมมติว่าข้อมูล อาจารย์คณะวิทยาศาสตร์มี 150 ดังนี้

เพศ (sex) : ชาย = 1 , หญิง = 2

อายุ (age) : ? ปี

สถานภาพสมรส (status) : โสด = 1 , แต่งงาน = 2 , อื่นๆ = 3

ปริญญาสูง (degree) : ตรี = 1 , โท = 2 , เอก = 3

จำนวนปีที่สอน (experience) : ? ปี

จงเขียน

1. โปรแกรมรับข้อมูลของอาจารย์คณะวิทยาศาสตร์จำนวน 150 คน
2. หาจำนวนปีเฉลี่ยของ อายุ และ จำนวนปีที่สอน จำนวนคนที่มีค่าอายุและประสบการณ์มากกว่าค่าเฉลี่ย

AVEREAGE AGE = xx.xx Years

NUMBER OF AGE ABOVE AVEAGE = XX

AVEREAGE Experience = XX.XX Years

NUMBER OF AGE ABOVE AVEAGE = XX

AvgStats.pas

```
1 program AverageStats;
2
3 const
4     numPerson = 3;
5 var
6     sex, age, stat, degr, tyear: array[1..numPerson] of Integer;
7     i: Integer;
8     avg_age, avg_expr: Real;
9     numAgeAbove, numYearAbove: Integer;
10 begin
11     { Input data }
12     for i := 1 to numPerson do
13     begin
14         Write('sex = '); ReadLn(sex[i]);
15         Write('age = '); ReadLn(age[i]);
16         Write('stat = '); ReadLn(stat[i]);
17         Write('degr = '); ReadLn(degr[i]);
18         Write('tyear = '); ReadLn(tyear[i]);
19     end;
20
```

AvgStats.pas

```
21  avg_age := 0;
22  avg_expr := 0;
23
24  { Compute sums }
25  for i := 1 to numPerson do
26  begin
27      avg_age := avg_age + age[i];
28      avg_expr := avg_expr + tyear[i];
29  end;
30
31  { Compute averages }
32  avg_age := avg_age / numPerson;
33  avg_expr := avg_expr / numPerson;
```

AvgStats.pas

```
37
38 { Count how many are above average }
39 for i := 1 to numPerson do
40 begin
41     if age[i] > avg_age then
42         numAgeAbove := numAgeAbove + 1;
43
44     if tyear[i] > avg_expr then
45         numYearAbove := numYearAbove + 1;
46 end;
47
48 { Output results }
49 WriteLn('AVG AGE = ', avg_age:0:2);
50 WriteLn('NUMBER OF AGE ABOVE AVERAGE = ', numAgeAbove);
51 WriteLn('AVG EXPR = ', avg_expr:0:2);
52 WriteLn('NUMBER OF YEAR ABOVE AVERAGE = ', numYearAbove);
53 end.
54
```

What are the problems of the above program?

Why we use const : numPerson ?

AvgStats.pas

```
main.pas F8
1 program AverageStats;
2
3 const
4   numPerson = 3;
5 var
6   sex, age, stat, degr, tyear: array[1..numPerson] of Integer;
7   i: Integer;
8   avg_age, avg_expr: Real;
9   numAgeAbove, numYearAbove: Integer;
10 begin
11   { Input data }
12   for i := 1 to numPerson do
13   begin
14     Write('sex = '); ReadLn(sex[i]);
15     Write('age = '); ReadLn(age[i]);
16     Write('stat = '); ReadLn(stat[i]);
17     Write('degr = '); ReadLn(degr[i]);
18     Write('tyear = '); ReadLn(tyear[i]);
19   end;
20
21   avg_age := 0;
22   avg_expr := 0;
23
24   { Compute sums }
25   for i := 1 to numPerson do
26   begin
27     avg_age := avg_age + age[i];
28     avg_expr := avg_expr + tyear[i];
29   end;
30
31   { Compute averages }
32   avg_age := avg_age / numPerson;
33   avg_expr := avg_expr / numPerson;
```

```
34
35   numAgeAbove := 0;
36   numYearAbove := 0;
37
38   { Count how many are above average }
39   for i := 1 to numPerson do
40   begin
41     if age[i] > avg_age then
42       numAgeAbove := numAgeAbove + 1;
43
44     if tyear[i] > avg_expr then
45       numYearAbove := numYearAbove + 1;
46   end;
47
48   { Output results }
49   WriteLn('AVG AGE = ', avg_age:0:2);
50   WriteLn('NUMBER OF AGE ABOVE AVERAGE = ', numAgeAbove);
51   WriteLn('AVG EXPR = ', avg_expr:0:2);
52   WriteLn('NUMBER OF YEAR ABOVE AVERAGE = ', numYearAbove);
53 end.
```

- Record (Structure) คือ กลุ่มของรายการข้อมูลที่เกี่ยวข้องกัน

ฟิลด์						
รหัสผู้ใช้	ชื่อ - นามสกุล	เพศ	ที่อยู่	เบอร์ติดต่อ	วันเกิด	เรคอร์ด
B00514	สุวารี ใจสบาย	ญ	34 คลองสาน	025478556	12/01/1977	
A00128	กนก ยินดี	ช	126 บางพลัด	026441255	09/07/1986	
A01245	เพ็ญศรี สุขสม	ญ	49/2 จตุจักร	014544255	05/11/1971	
C13244	บุญมี ยินดี	ช	66 ลาดพร้าว	066458554	26/04/1980	
.

11.3. Record/ Structure

- In a database program , data is stored in records

records

Item ID : ABC-456-299

Description : Remote Control Monster Truck

Quantity On Hand : 9

Cost : 47.80 Retail Price : 98

field

11.3. Record/Structure

- คือ ชนิดข้อมูลแบบกำหนดเอง (**user-defined data type**) ที่ใช้ที่ข้อมูลที่มีความสัมพันธ์กันหลาย ๆ ค่าไว้ด้วยกัน ภายใต้ตัวแปรเดียว โดยข้อมูลเหล่านั้นอาจเป็น คนละชนิด (**data type**) กันได้

```
2 program recordDeclare;
3 type
4   InventoryItem = record
5     item_id: string[30];
6     description: string[50];
7     OnHand: Integer;
8     cost: real;
9     retail_price: real;
10  end;
11 var
12   item: InventoryItem;
13 begin
14   item.item_id := 'A1001';
15   item.description := 'Wireless Mouse';
16   item.OnHand := 25;
17   item.cost := 10.5;
18   item.retail_price := 15.9;
19 end.
20
```

It's just a user defined datatype
(variable template).



Define structure type

type

name = record

member-list (fields_name)

end;

ตัวอย่าง

type

PERSON = record

age:integer;

tell:integer;

weight:real;

name:string[10];

end;

ความหมายของข้อมูล

-ข้อมูลของคน (person) ที่ระบบจัดเก็บ
ประกอบด้วย

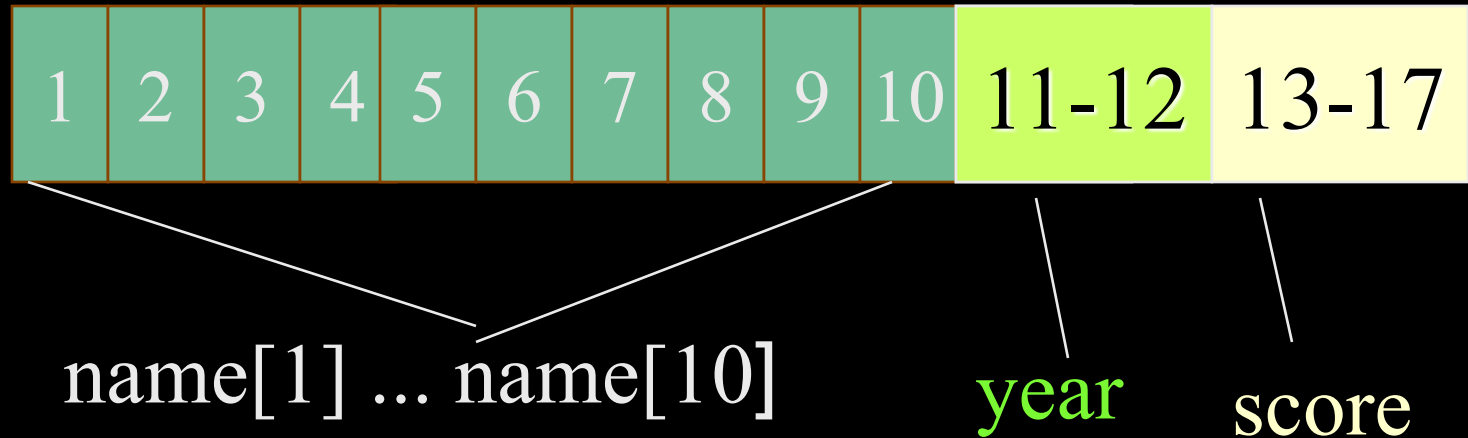
อายุ (age), ความสูง(tell),

น้ำหนัก (weight) ,ชื่อ (name)

2. ตัวอย่างการสร้าง template

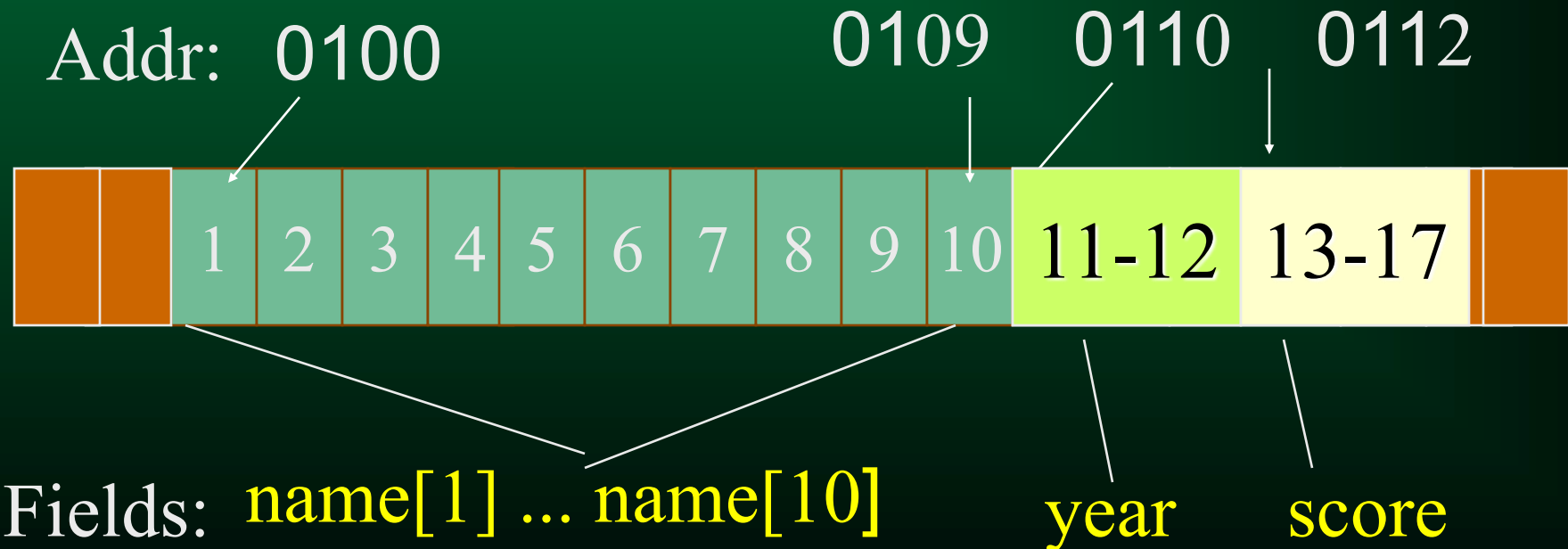
```
1 program examplePersion;  
2  
3 type  
4 StudentRec = record  
5     name: string[10];  
6     year : Integer;  
7     score : Real;  
8 end;  
9
```

Suppose int (2 bytes)
long ,float (4 bytes)
char (1) bytes



Declare var

```
1  program examplePersion;  
2  
3  type  
4    StudentRec = record  
5      name: string[10];  
6      year : Integer;  
7      score : Real;  
8    end;  
9  
10 var  
11   Somsri : StudentRec;  
12
```





การอ้างข้อมูลใน **field** ของตัวแปร **structure/record**

referencing data via pointer

var_name.fields

(variable)

var_ary [index].fields

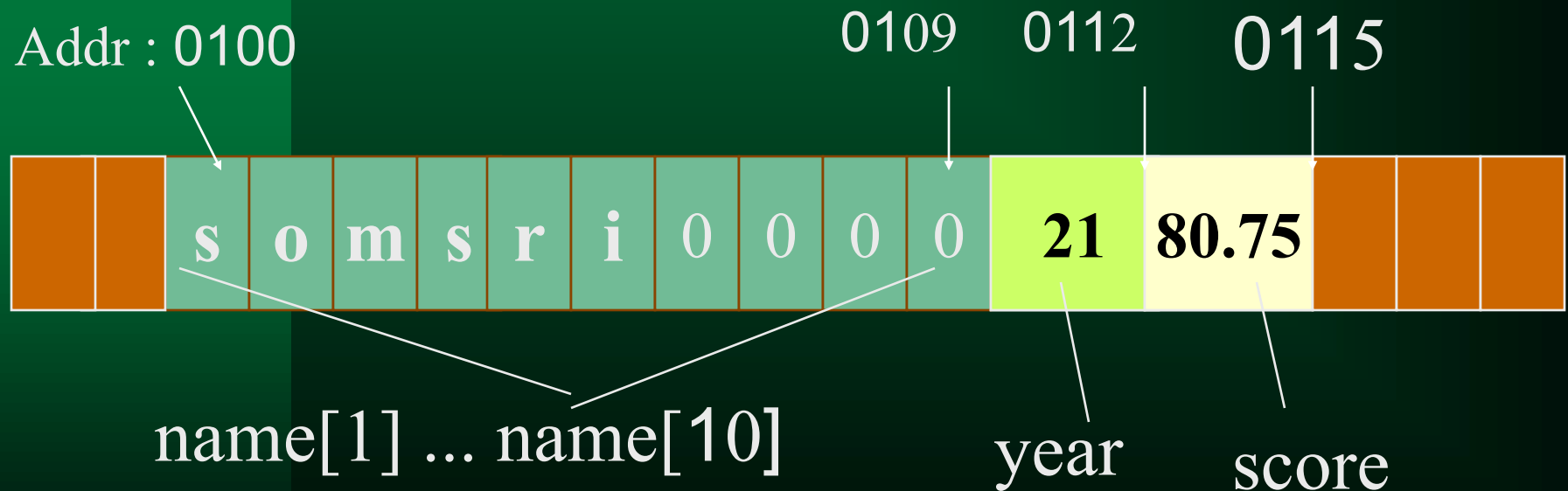
(array index)

Structure : การอ้างข้อมูล

var_name . filed_name

```
13 begin
14   Somsri.name := 'Somsri';
15   Somsri.year  := 21;
16   Somsri.score := 80.75;
17
18   writeln( 'sizeof(Somsri)=', sizeof(Somsri) );
19   writeln( 'Somsri.name=', Somsri.name);
20   writeln( 'Somsri.year=', Somsri.year);
21   writeln( 'Somsri.score=', Somsri.score:0:2);
22
23 end.
```

Suppose int (2 bytes)
long ,float (4 bytes)
char (1) bytes



main.pas

```
1 program examplePersion;
2
3 type
4   StudentRec = record
5     name: string[10];
6     year : Integer;
7     score : Real;
8   end;
9
10 var
11   Somsri : StudentRec;           { individual v
12
13 begin
14   Somsri.name := 'Somsri';
15   Somsri.year := 21;
16   Somsri.score := 80.75;
17
18   writeln( 'sizeof(Somsri)=', sizeof(Somsri) );
19   writeln( 'Somsri.name=', Somsri.name);
20   writeln( 'Somsri.year=', Somsri.year);
21   writeln( 'Somsri.score=', Somsri.score:0:2);
22
23 end.
24
```

```
sizeof(Somsri)=24
Somsri.name=Somsri
Somsri.year=21
Somsri.score=80.75
```


Operation ที่เกี่ยวข้องกับ Structure [1]

```
3 type
4   StudentRec = record
5     name: string[10];
6     year : Integer;
7     score : Real;
8   end;
9
10  PERSON  = record
11    age:integer;
12    tell:integer;
13    weight:real;
14    name:string[20];
15  end;
16
17 var
18   Somsri,tip : StudentRec;           { individual variables }
19   one ,two :PERSON;
```

Operation ที่เกี่ยวข้องกับ Structure [2]

1. การ assign ระดับ structure level

tip := Somsri ได้

two := one ได้

tip = one ไม่ได้

2. ไม่สามารถใช้ + , - , * , % , < , > , >= , <= ใน structure level ได้

one = one + two

Somsri > tip

one < Somsri

```
3 type
4   StudentRec = record
5     name: string[10];
6     year : Integer;
7     score : Real;
8   end;
9
10  PERSON = record
11    age:integer;
12    tell:integer;
13    weight:real;
14    name:string[20];
15  end;
16
17  var
18    Somsri,tip : StudentRec;
19    one ,two :PERSON;
```

Operation ที่เกี่ยวข้องกับ Structure [3]

1. การ assign

one.weight := 50.5;

two.name := Somsri.name;

two.name = one.age

ได้

ได้

ไม่ได้

2. ใช้ +, -, *, %, <, >, >=, <= ได้

two.age := one.age + Somsri.year;

two.name := Somsri.name;

Somsri.year < two.tell

somjai.name < somsri.age

ได้

ได้

ได้

ไม่ได้

```

20 begin
21     Somsri.name := 'Somsri';
22     Somsri.year := 21;
23     Somsri.score := 80.75;
24     tip := Somsri;
25     tip.year := Somsri.year + 12;
26
27     writeln( 'tip.year <> Somsri.year ->',
28             tip.year <> Somsri.year );
29     //writeln( 'tip <> Somsri', tip <> Somsri );
30     one.name := 'one';
31     one.age := 22;
32     one.tell := 170;
33     one.weight := 50.5;
34     two := one;
35     two.age := one.age+Somsri.year;
36     writeln( 'one.age < two.tell ->', one.age < two.tell );
37     writeln( 'Somsri.year < two.tell ->', Somsri.year < two.tell );
38
39     //tip := one;
40     //one := one + two;
41
42     two.name := Somsri.name;
43     writeln( 'Somsri.name = ->', Somsri.name );
44     writeln( 'one.name = ->', two.name );
45
46     one.name := two.name;
47     writeln( 'one.name = ->', one.name );
48
49
50 end.

```

```

1 program examplePersion;
2
3 type
4     StudentRec = record
5         name: string[10];
6         year : Integer;
7         score : Real;
8     end;
9
10    PERSON = record
11        age:integer;
12        tell:integer;
13        weight:real;
14        name:string[20];
15    end;
16
17 var
18     Somsri,tip : StudentRec;
19     one ,two :PERSON;

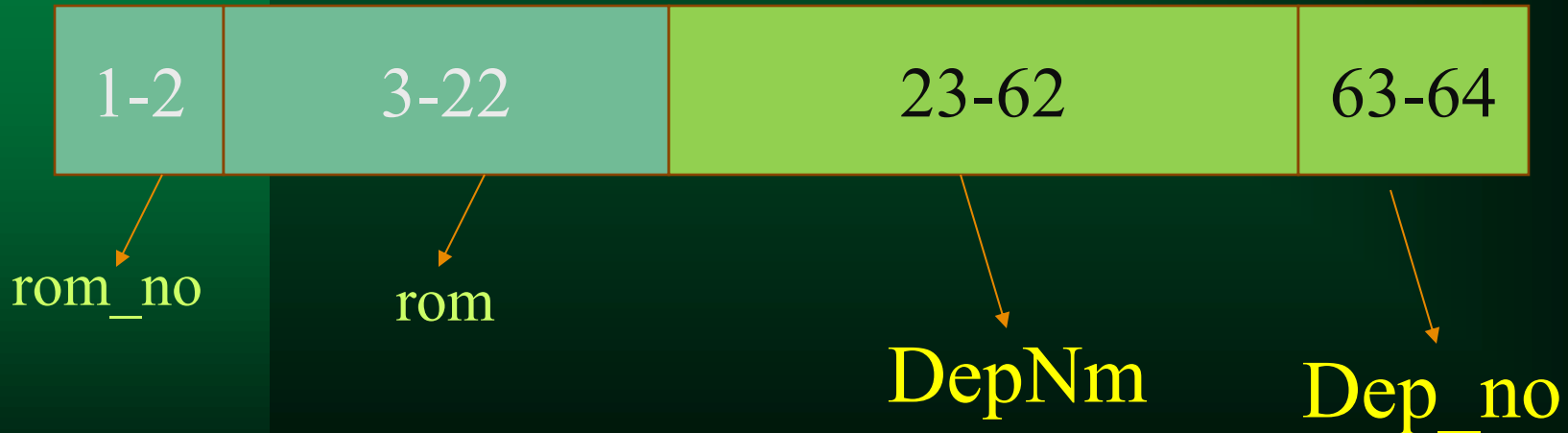
```

การสร้าง sub-record/structure

Suppose int (2 bytes)
long ,float (4 bytes)
char (1) bytes

```
3 type
4   DEPARTMENT = record
5     DepNm: string[40];
6     Dep_no: Integer;
7   end;
8
9   ROOM = record
10    rom_no: Integer;
11    Rom: string[20];
12    Dept: DEPARTMENT; { nested record }
13  end;
14
15 var
16   r1: ROOM;
```

Byte order :



```

1 program ExampleDepartmentRoom;
2
3 type
4     DEPARTMENT = record
5         DepNm: string[40];
6         Dep_no: Integer;
7     end;
8
9     ROOM = record
10        rom_no: Integer;
11        Rom: string[20];
12        Dept: DEPARTMENT;    { nested record }
13    end;
14
15 var
16     r1: ROOM;
17 begin
18     { Assign department info }
19     r1.Dept.DepNm := 'Computer Science';
20     r1.Dept.Dep_no := 101;
21
22     { Assign room info }
23     r1.rom_no := 12;
24     r1.Rom := 'Lab A';
25
26     { Display output }
27     writeln('Room No: ', r1.rom_no);
28     writeln('Room Name: ', r1.Rom);
29     writeln('Department Name: ', r1.Dept.DepNm);
30     writeln('Department No: ', r1.Dept.Dep_no);
31 end.
32

```

DepRoom.pas

Example of structure variable

nain.pas

```
1  program examplePersion;
2
3  type
4  PERSON = record
5      age: Integer;
6      tell: Integer;
7      weight: Real;
8      name: string[10];
9  end;
10
11 DEPARTMENT = record
12     nID: Integer;
13     DepNm: string[80];
14     student : array[1..30] of PERSON;
15 end;
16
17 var
18     dang, pin, tip: PERSON;           { individual variables }
19     family: array[1..5] of PERSON;    { array of PERSON }
20
21     science: DEPARTMENT;             { individual variable }
22     ramFaculty: array[1..6] of DEPARTMENT; { array of DEPARTMENT }
```

exPersion.pas

Example create structure and its variables

exPersion.pas

```
24 begin
25     dang.age := 31;
26     dang.tell := 170;
27     dang.weight := 45.6;
28     dang.name := 'dang';
29
30     tip := dang;           //copy
31     tip.name := 'tip';
32
33     family[1] := dang;     //copy
34     family[2] := tip;     //copy
35
36     science.niD := 1;
37     science.DepNm := 'science';
38     science.student[1] := dang;
39     science.student[2] := tip;
40
41     ramFaculty[1] := science; //copy
42
```


Example create structure and its variables

```
43  writeln( 'sizeof(dang)=', sizeof(dang),
44          ',dang.name=',dang.name ,',dang.tell=',dang.tell );
45
46  writeln( 'sizeof(family)=', sizeof(family),
47          ',family[1].name=',family[1].name ,
48          ',family[1].tell=',family[1].tell );
49
50  writeln( 'sizeof(science)=', sizeof(science), |
51          ',science.student[1].name=',science.student[1].name ,
52          ',science.student[1].tell=',science.student[1].tell );
53
54  writeln( 'sizeof(ramFaculty)=', sizeof(ramFaculty),
55          ',ramFaculty[1].student[1].name=',ramFaculty[1].student[1].name ,
56          ',ramFaculty[1].student[1].tell=',ramFaculty[1].student[1].tell );
57  end.
58
sizeof(dang)=32 ,dang.name=dang,dang.tell=170
sizeof(family)=160 ,family[1].name=dang,family[1].tell=170
sizeof(science)=1048 ,science.student[1].name=dang,science.student[1].tell=170
sizeof(ramFaculty)=6288 ,ramFaculty[1].student[1].name=dang,ramFaculty[1].student[1].tell=170
```

exPersion.pas



แบบฝึกหัด : ให้นักศึกษาสร้าง structure
สำหรับการเก็บข้อมูลต่อไปนี้

สมมติว่าต้องการเก็บข้อมูลนักศึกษา 2 คน ประกอบด้วยรายการข้อมูลดังนี้

รหัสประจำตัว	ชื่อ	ที่อยู่	เพศ	อายุ	เกรดเฉลี่ย
61001	Dang	Bangkok	0	19	2.50
61002	Sri	Bangkok	1	20	2.75

การประกาศระเบียบนภายในระเบียบน

สมมติว่าต้องการเก็บข้อมูลนักศึกษา ประกอบด้วยรายการข้อมูลดังนี้

รหัสประจำตัว	ชื่อ	ที่อยู่	เพศ	อายุ	เกรดเฉลี่ย	วันเกิด	เดือนเกิด	ปีเกิด
61001	Dang	Bangkok	0	19	2.50	25	11	2540

type

```
Student = record
  code: string[10];
  name: string[30];
  addr: string[100];
  sex: Integer;
  age: Integer;
  avg_grade: Real;
  dob_day: Integer;
  dob_month: Integer;
  dob_year: Integer;
end;
```

type

```
DateRec = record
  day: Integer;
  month: Integer;
  year: Integer;
end;

Student = record
  code: string[10];
  name: string[30];
  addr: string[100];
  sex: Integer;
  age: Integer;
  avg_grade: Real;
  dob: DateRec;
end;
```

HOMework1 : จากตัวอย่างที่ 1 ให้ นศแปลงโปรแกรมโดยใช้เทคนิค structure

สมมติว่าข้อมูล อาจารย์คณะวิทยาศาสตร์มี 150 ดังนี้

เพศ (sex) : ชาย = 1 , หญิง = 2

อายุ (age) : ? ปี

สถานภาพสมรส (status) : โสด = 1 , แต่งงาน = 2 , อื่นๆ = 2

ปริญญาสูง (degree) : ตรี = 1 , โท = 2 , เอก = 3

จำนวนปีที่สอน (experience) : ? ปี

จงเขียน

1. โปรแกรมรับข้อมูลของอาจารย์คณะวิทยาศาสตร์
2. หาจำนวนปีเฉลี่ยของ อายุ และ จำนวนปีที่สอน

AVERAGE AGE = xx.xx Years

AVERAGE Experience = XX.XX Years

main.pas

```

1 program EmployeeAverage;
2 const numPerson = 3; //ค่าคงที่
3 type
4   Employee = record
5     sex: Integer;
6     age: Integer;
7     stat: Integer;
8     degr: Integer;
9     tyear: Integer;
10  end;
11
12 var
13   person: array[1..numPerson] of Employee;
14   i: Integer;
15   avg_age, avg_expr: Real;
16   numAgeAbove, numYearAbove: Integer;
17

```

StruEmpAvg.pas

HelloWorld.pas

44

```

1 program AverageStats;
2
3 const
4   numPerson = 3;
5 var
6   sex, age, stat, degr, tyear: array[1..numPerson] of Integer;
7   i: Integer;
8   avg_age, avg_expr: Real;
9   numAgeAbove, numYearAbove: Integer;

```

StruEmpAvg.pas

```
17
18 begin
19   { Input data for first 2 employees }
20   for i := 1 to numPerson do
21   begin
22     Write('sex(1-2) = '); ReadLn(person[i].sex);
23     Write('age = '); ReadLn(person[i].age);
24     Write('stat (1-3) = '); ReadLn(person[i].stat);
25     Write('degr(1-3) = '); ReadLn(person[i].degr);
26     Write('tyear = '); ReadLn(person[i].tyear);
27   end;
```

```
10 begin
11   { Input data }
12   for i := 1 to numPerson do
13   begin
14     Write('sex = '); ReadLn(sex[i]);
15     Write('age = '); ReadLn(age[i]);
16     Write('stat = '); ReadLn(stat[i]);
17     Write('degr = '); ReadLn(degr[i]);
18     Write('tyear = '); ReadLn(tyear[i]);
19   end;
```

StruEmpAvg.pas

```
28
29 avg_age := 0;
30 avg_expr := 0;
31
32 { Compute sums }
33 for i := 1 to numPerson do
34 begin
35     avg_age := avg_age + person[i].age;
36     avg_expr := avg_expr + person[i].tyear;
37 end;
38
39 avg_age := avg_age / numPerson;
40 avg_expr := avg_expr / numPerson;
```

```
20
21 avg_age := 0;
22 avg_expr := 0;
23
24 { Compute sums }
25 for i := 1 to numPerson do
26 begin
27     avg_age := avg_age + age[i];
28     avg_expr := avg_expr + tyear[i];
29 end;
30
31 { Compute averages }
32 avg_age := avg_age / numPerson;
33 avg_expr := avg_expr / numPerson;
34
```

StruEmpAvg.pas

```
41 numAgeAbove := 0;  
42 numYearAbove := 0;  
43  
44 { Count how many are above average }  
45 for i := 1 to numPerson do  
46 begin  
47     if person[i].age > avg_age then  
48         numAgeAbove := numAgeAbove + 1;  
49  
50     if person[i].tyear > avg_expr then  
51         numYearAbove := numYearAbove + 1;  
52     end;  
53
```

```
37  
38 { Count how many are above average }  
39 for i := 1 to numPerson do  
40 begin  
41     if age[i] > avg_age then  
42         numAgeAbove := numAgeAbove + 1;  
43  
44     if tyear[i] > avg_expr then  
45         numYearAbove := numYearAbove + 1;  
46     end;  
47
```


StruEmpAvg.pas

```
54  
55   { Output results }  
56   WriteLn('AVG AGE = ', avg_age:0:2);  
57   WriteLn('NUMBER OF AGE ABOVE AVERAGE = ', numAgeAbove);  
58   WriteLn('AVG EXPR = ', avg_expr:0:2);  
59   WriteLn('NUMBER OF YEAR ABOVE AVERAGE = ', numYearAbove);  
60   end.  
61
```

```
47  
48   { Output results }  
49   WriteLn('AVG AGE = ', avg_age:0:2);  
50   WriteLn('NUMBER OF AGE ABOVE AVERAGE = ', numAgeAbove);  
51   WriteLn('AVG EXPR = ', avg_expr:0:2);  
52   WriteLn('NUMBER OF YEAR ABOVE AVERAGE = ', numYearAbove);  
53   end.  
54
```

StruEmpAvg.pas

main.pas

```
1 program EmployeeAverage;
2 const numPerson = 3; //ค่าคงที่
3 type
4   Employee = record
5     sex: Integer;
6     age: Integer;
7     stat: Integer;
8     degr: Integer;
9     tyear: Integer;
10  end;
11
12 var
13   person: array[1..numPerson] of Employee;
14   i: Integer;
15   avg_age, avg_expr: Real;
16   numAgeAbove, numYearAbove: Integer;
17
18 begin
19   { Input data for first 2 employees }
20   for i := 1 to numPerson do
21     begin
22       Write('sex(1-2) = '); ReadLn(person[i].sex);
23       Write('age = '); ReadLn(person[i].age);
24       Write('stat (1-3) = '); ReadLn(person[i].stat);
25       Write('degr(1-3) = '); ReadLn(person[i].degr);
26       Write('tyear = '); ReadLn(person[i].tyear);
27     end;
28
29   avg_age := 0;
30   avg_expr := 0;
```

```
29   avg_age := 0;
30   avg_expr := 0;
31
32   { Compute sums }
33   for i := 1 to numPerson do
34     begin
35       avg_age := avg_age + person[i].age;
36       avg_expr := avg_expr + person[i].tyear;
37     end;
38
39   avg_age := avg_age / numPerson;
40   avg_expr := avg_expr / numPerson;
41
42   numAgeAbove := 0;
43   numYearAbove := 0;
44
45   { Count how many are above average }
46   for i := 1 to numPerson do
47     begin
48       if person[i].age > avg_age then
49         numAgeAbove := numAgeAbove + 1;
50
51       if person[i].tyear > avg_expr then
52         numYearAbove := numYearAbove + 1;
53     end;
54
55   { Output results }
56   WriteLn('AVG AGE = ', avg_age:0:2);
57   WriteLn('NUMBER OF AGE ABOVE AVERAGE = ', numAgeAbove);
58   WriteLn('AVG EXPR = ', avg_expr:0:2);
59   WriteLn('NUMBER OF YEAR ABOVE AVERAGE = ', numYearAbove);
60 end.
```

HOMEWORK2 : จาก HOMEWORK1 ให้ นักศึกษาเขียนโปรแกรมคำนวณ

- 1.ค่าเฉลี่ยของอายุจำแนกตาม เพศ
- 2.นับจำนวนคนจำแนกตาม เพศ และปริญญา
3. จำนวนปีที่สอนเฉลี่ยจำแนกตาม เพศ
4. หาจำนวนปีเฉลี่ยของ อายุ และ จำนวนปีที่สอน จำนวนคนที่มีค่าอายุและประสบการณ์มากกว่าค่าเฉลี่ย

AVEREAGE AGE = xx.xx Years

NUMBER OF AGE ABOVE AVEAGE = XX

AVEREAGE Experience = XX.XX Years

NUMBER OF AGE ABOVE AVEAGE = XX

NUMBER OF Men SEX ABOVE AVEAGE = XX

NUMBER OF Women SEX ABOVE AVEAGE = XX

Ex 1 จงใช้โครงสร้างข้อมูลแบบ record

น.ศ. 50 คน แต่ละคนสอบ 3 ครั้งในวิชาเขียนโปรแกรม P1,P2,P3 และได้คะแนนสอบ 3 ครั้งในวิชาคณิตศาสตร์ (M1,M2,M3)

จงเขียนโปรแกรมรับ ชื่อ ,คะแนน,ของแต่ละวิชา ในแต่ละครั้ง เพื่อหาคะแนนเฉลี่ยของ น.ศ. แต่ละคนแต่ละวิชา รวมทั้งคนที่มีคะแนนเฉลี่ยวิชาคณิตศาสตร์สูงสุด

{ 20 30 20
40 50 20

คณิตศาสตร์

เขียนโปรแกรม

ExScore.pas

น.ศ. คนที่ { 35 30 25
32 20 50

คณิตศาสตร์

เขียนโปรแกรม

```

1 program ExampleScore;
2
3 type
4   Score = record
5     name: string[30];
6     P1, P2, P3: Integer;
7     M1, M2, M3: Integer;
8     avgP, avgM: Real;
9   end;
10
11 var
12   std: array[1..50] of Score;
13   i, imax: Integer;
14   avg_max: Real;
15

```

```

16 begin
17   { Input data }
18   for i := 1 to 50 do
19     begin
20       write('Enter name of student ', i, ': ');
21       readln(std[i].name);
22       write('Enter P1, P2, P3: ');
23       readln(std[i].P1, std[i].P2, std[i].P3);
24       write('Enter M1, M2, M3: ');
25       readln(std[i].M1, std[i].M2, std[i].M3);
26     end;

```

ExScore.pas

```

27
28 { calculate averages }
29 for i := 1 to 50 do
30 begin
31     std[i].avgP := (std[i].P1 + std[i].P2 + std[i].P3) / 3.0;
32     std[i].avgM := (std[i].M1 + std[i].M2 + std[i].M3) / 3.0;
33 end;
34
35 { Find max avgM }
36 avg_max := std[1].avgM;
37 imax := 1;
38 for i := 2 to 50 do
39 begin
40     if avg_max < std[i].avgM then
41     begin
42         avg_max := std[i].avgM;
43         imax := i;
44     end;
45 end;

```

ExScore.pas

```

46
47 writeln;
48 writeln('Max name = ', std[imax].name);
49 writeln('Max avgM = ', avg_max:0:2);
50 end.

```



ตัวอย่าง

ให้นักศึกษาเขียนโปรแกรมขายสินค้าและพิมพ์ใบเสร็จรับเงิน
โดยข้อมูลในระบบมีไม่เกิน 100 รายการ

ข้อมูลขายสินค้าด้วย

- รหัสสินค้า(id) เช่น “AB001”
- ชื่อสินค้า(name) เช่น “CAFÉ”
- ราคาขาย(cost) เช่น 30.6
- จำนวนสินค้าในสต็อก (instock) เช่น 5

prodMgr.pas

Menu

1. Add item to Cart
2. Sell Product
3. Check Stock
4. End Program

Please select 1 / 2 / 3/ 4 =?

Menu

1. Add item to Cart
2. Sell Product
3. Check Stock
4. End Program

Please select 1 / 2 / 3/ 4 =?

กด 1

Input your product data:

id : AB001

name : CAFE

Cost : 30.6

Stock : 5

กด 2

Input your product id: AB001

Input your sell volume : 2

name : CAFE

Cost : 30.6

stock : 3

Pay : 61.2 bath

Apirak :

Menu

1. Add item to Cart
2. Sell Product
3. Check Stock
4. End Program

Please select 1 / 2 / 3/ 4 =?

กด 3

product List:

Id	name	cost	stock
----	------	------	-------

AB001	CAFÉ	30.6	3
-------	------	------	---

กด 4

exit from program

Menu

1. Add item to Cart
2. Sell product
3. Check stock
4. End program

Please select 1 / 2 / 3/ 4 =? 1

New product:

id =? a001

name=? cafel

cost=? 50

instock=? 20

Menu

1. Add item to Cart
2. Sell product
3. Check stock
4. End program

Please select 1 / 2 / 3/ 4 =? 1

New product:

id =? a002

name=? cafel2345

cost=? 45.5

instock=? 10

เมนู

กด 1

เมนู

กด 1

instock=? 10

Menu

1. Add item to Cart
2. Sell product
3. Check stock
4. End program

Please select 1 / 2 / 3/ 4 =? 3

id	name	cost	instock
a001	cafe1	50	20
a002	cafe12345	45.5	10

Menu

1. Add item to Cart
2. Sell product
3. Check stock
4. End program

Please select 1 / 2 / 3/ 4 =? 2

Input your product id: a001

Input your buy volumn: 2

name= cafe1

cost=50

instock=20

Pay = 100 Menu

1. Add item to Cart
2. Sell product
3. Check stock
4. End program

Please select 1 / 2 / 3/ 4 =? 3

id	name	cost	instock
a001	cafe1	50	18
a002	cafe12345	45.5	10

Menu

1. Add item to Cart
2. Sell product

กด 3

กด 2

กด 3

```
1  program ProductManagement;
2
3  type
4  TProduct = record
5      id: string[10];
6      name: string[30];
7      cost: Real;
8      instock: Integer;
9  end;
10
11 var
12     item: array[1..100] of TProduct;
13     n, select, i, buyvolume: Integer;
14     id: string[30];
15     found: Boolean;
16
17 begin
18     n := 0;
19     while True do
20     begin
21         WriteLn(' Menu');
22         WriteLn('1. Add item to Cart');
23         WriteLn('2. Sell product');
24         WriteLn('3. Check stock');
25         WriteLn('4. End program');
26         Write('Please select 1 / 2 / 3 / 4 =? ');
27         ReadLn(select);
28     end;
```

Draft code (1)

prodMgr.pas

it code (1)

```
31
32 case select of
33   1: begin
34     WriteLn('New product:');
35     Write('id=? '); ReadLn(item[n + 1].id);
36     Write('name=? '); ReadLn(item[n + 1].name);
37     Write('cost=? '); ReadLn(item[n + 1].cost);
38     Write('instock=? '); ReadLn(item[n + 1].instock);
39     n := n + 1;
40   end;
41
42   2: begin
43     Write('Input your product id: ');
44     ReadLn(id);
45     Write('Input your buy volume: ');
46     ReadLn(buyvolume);
47     found := False;
48
49     for i := 1 to n do
50       begin
51         if item[i].id = id then
52           begin
53             found := True;
54             Break;
55           end;
56       end;
57
58     if found then
59       begin
60         WriteLn('name = ', item[i].name);
61         WriteLn('cost = ', item[i].cost:0:2);
62         WriteLn('instock = ', item[i].instock);
63         WriteLn('Pay = ', (item[i].cost * buyvolume):0:2);
64         item[i].instock := item[i].instock - buyvolume;
65       end
66     else
67       WriteLn('Product not found!');
68     end;
69
```

prodMgr.pas

Draft code (1)

prodMgr.pas

```
69
70 3: begin
71     WriteLn;
72     WriteLn( 'id',#9, 'name',#9, 'cost',#9, 'instock');
73     for i := 1 to n do
74     begin
75         writeln(item[i].id,#9, item[i].name,#9,
76                item[i].cost:0:2,#9, item[i].instock);
77     end;
78 end;
79 end;
80
81 end; //while True do
82
83 end.
84
```

EX1

จากตัวอย่างข้างต้นให้ นักศึกษาปรับโปรแกรม โดยให้มีการ
ตรวจดังนี้

กรณี กค 1 : ให้ตรวจสอบว่ารหัสที่ป้อน มีซ้ำหรือไม่ ถ้าไม่มีจึงจะ
อนุญาตให้ผู้ใช้ป้อนข้อมูล

กรณี กค 2: ให้ตรวจสอบว่ารหัสสินค้าที่ผู้ใช้ป้อนมีอยู่ในระบบ
หรือไม่ กรณีไม่พบให้แสดงข้อความ “product not exist”

กรณี กค 3 ให้แสดงรายการทั้งหมด และแสดงยอดรวมของ cost

SeatResv.pas

product List:

Id	name	cost	stock
----	------	------	-------

-------	--	--	--

AB001	CAFÉ	30.6	3
-------	------	------	---

ปัญหา ต้องการจองที่นั่งเพื่อไปทัศนศึกษาจังหวัดสุโขทัย แผนผังของที่นั่งมี 8 แถว แต่ละแถวมี 4 ที่นั่ง กรณีที่นั่งถูกจองแล้วจะไม่อนุญาตให้จองซ้ำ

1A	1B	ทางเดิน	1C	1D
2A	2B		2C	2D
3A	3B		3C	3D
4A	4B		4C	4D
5A	5B		5C	5D
6A	6B		6C	6D
7A	7B		7C	7D
8A	8B		8C	8D

3. สร้างเมนูเพื่อให้ผู้ใช้เลือกทำงาน โดยการทำงานจะกระทำซ้ำ

กรณี กด “R” จอที่ว่าง

โปรแกรมจะแสดงที่ว่างเพื่อให้ผู้ใช้เลือก เพื่อ

ป้อน ชื่อ(name) เบอร์โทรศัพท์(tel) ที่ว่าง

(number) และ เปลี่ยนสถานะที่ว่าง (free) ให้มี

ค่าเป็น false

กด “S” แสดงรายงานการจองที่ว่าง ได้แก่ ชื่อ ,เบอร์โทร,

กด “Q” จบการทำงาน หมายเลขแก้ไข



วิเคราะห์

วิเคราะห์ข้อมูลในส่วนของ

- ข้อมูลนำเข้า
- ผลลัพธ์ที่ต้องการ และ
- การประมวลผล

ผลลัพธ์ที่ต้องการ

เริ่มต้นโปรแกรมแสดง
ที่นั่ง และ เมนู

เลือกจอง 2B

โปรแกรมแสดงที่นั่งและเมนู

1A	1B	1C	1D
2A	2B	2C	2D
3A	3B	3C	3D
4A	4B	4C	4D
5A	5B	5C	5D
6A	6B	6C	6D
7A	7B	7C	7D
8A	8B	8C	8D

MENU

<R> Reserve seat
 <S> Summary seat
 <Q> Quit

select R/S/Q ? R

select seat (upper charector) : 2B

name=? urai

telephone=? 123456

1A	1B	1C	1D
2A	**	2C	2D
3A	3B	3C	3D
4A	4B	4C	4D
5A	5B	5C	5D
6A	6B	6C	6D
7A	7B	7C	7D
8A	8B	8C	8D

MENU

<R> Reserve seat
 <S> Summary seat
 <Q> Quit

select R/S/Q ?



ข้อมูลนำเข้า

ข้อมูลนำเข้า

ประกอบด้วย

ชื่อ เบอร์โทรศัพท์ และ

ที่นั่งที่ต้องการ

1. ประกาศโครงสร้างระเบียนชื่อ **Seat** ประกอบด้วยรายการข้อมูลดังนี้

- ชื่อ(name)
- เบอร์โทรศัพท์(tel)
- ที่นั่ง(number)

และเพิ่มรายการข้อมูลเพิ่มอีกรายการหนึ่งคือ

- สถานะที่นั่ง(free)

กรณีว่าง free มีค่าเป็น true

กรณีไม่ว่างมีค่าเป็น false

การประมวลผล

main.pas

```

1 program SeatReservation;
2
3 type
4   Seat = record
5     siteno: string[5];
6     free: boolean; // true = free, false = reserved
7     custname: string[50];
8     tel: string[10];
9   end;
10
11 var
12   site: array[1..8, 1..4] of Seat;
13   selectMenu: char;
14   row, col: integer;
15   iRfound, iCfound: integer;
16   selectSite: string[10];
17
18 begin
19   { Initialize seat data }
20   site[1,1].siteno := '1A'; site[1,2].siteno := '1B'; site[1,3].siteno := '1C'; site[1,4].siteno := '1D';
21   site[2,1].siteno := '2A'; site[2,2].siteno := '2B'; site[2,3].siteno := '2C'; site[2,4].siteno := '2D';
22   site[3,1].siteno := '3A'; site[3,2].siteno := '3B'; site[3,3].siteno := '3C'; site[3,4].siteno := '3D';
23   site[4,1].siteno := '4A'; site[4,2].siteno := '4B'; site[4,3].siteno := '4C'; site[4,4].siteno := '4D';
24   site[5,1].siteno := '5A'; site[5,2].siteno := '5B'; site[5,3].siteno := '5C'; site[5,4].siteno := '5D';
25   site[6,1].siteno := '6A'; site[6,2].siteno := '6B'; site[6,3].siteno := '6C'; site[6,4].siteno := '6D';
26   site[7,1].siteno := '7A'; site[7,2].siteno := '7B'; site[7,3].siteno := '7C'; site[7,4].siteno := '7D';
27   site[8,1].siteno := '8A'; site[8,2].siteno := '8B'; site[8,3].siteno := '8C'; site[8,4].siteno := '8D';
28
29   for row := 1 to 8 do
30     for col := 1 to 4 do
31       begin
32         site[row, col].free := True;
33         site[row, col].custname := '';
34         site[row, col].tel := '';
35       end;
36

```

SeatResv.pas

SeatResv.pas

```
36
37 repeat
38     writeln;
39     writeln('Current Seat Map:');
40     for row := 1 to 8 do
41     begin
42         for col := 1 to 4 do
43         begin
44             if site[row, col].free then
45                 write('|', site[row, col].sitenumber:3)
46             else
47                 write('|', ' **');
48         end;
49         writeln;
50     end;
51
52     writeln;
53     writeln('MENU');
54     writeln('<R> Reserve seat');
55     writeln('<S> Summary seat');
56     writeln('<Q> Quit');
57     write('Select R/S/Q ? ');
58     readln(selectMenu);
59
```

SeatResv.pas

```
59 case selectMenu of
60   'R', 'r':
61     begin
62       iRfound := -1;
63       iCfound := -1;
64       write('Select seat (Upper character): ');
65       readln(selectSite);
66
67       for row := 1 to 8 do
68         begin
69           for col := 1 to 4 do
70             begin
71               if site[row, col].sitenumber = selectSite then
72                 begin
73                   iRfound := row;
74                   iCfound := col;
75                   break;
76                 end;
77             end;
78           if iRfound <> -1 then break;
79         end;
80
81       if (iRfound <> -1) and (site[iRfound, iCfound].free) then
82         begin
83           site[iRfound, iCfound].free := False;
84           write('Name =? '); readln(site[iRfound, iCfound].custname);
85           write('Telephone =? '); readln(site[iRfound, iCfound].tel);
86         end
87       else
88         writeln('Seat not found or already reserved.');
```




SeatResv.pas

```
91
92   'S', 's':
93   begin
94       writeln;
95       writeln('Summary of Reserved Seats:');
96       for row := 1 to 8 do
97           for col := 1 to 4 do
98               if not site[row, col].free then
99                   writeln('Seat ', site[row, col].siteno, #9,
100                        'Name: ', site[row, col].custname, #9,
101                        'Tel: ', site[row, col].tel);
102           end;
103       end;
104
105   until (selectMenu = 'Q') or (selectMenu = 'q');
106 end.
107
```

Ex 2 จงใช้โครงสร้างข้อมูลแบบ record

เงินเดือนของพนักงาน XYZ ขึ้นอยู่กับ ฐานเงินเดือน ที่เท่ากันทุกคน คือ 3000 บาท บวกกับเปอร์เซ็นต์ของ ฐานเงินเดือนตาม ชนิดของงาน จำนวนปีที่ทำงาน ระดับความรู้ ความเอาใจใส่งาน

$$\text{รายได้ (ic)} = \text{เงินเดือน} + \text{ฐานเงินเดือน} * (\%$$

ชนิดของงาน	%	ระดับความรู้	%
1	0	1 (ป.ว.ช)	0
2	5	2 (ป.ว.ส)	5
3	15	3 (ป.ตรี)	15

Ex 2 จงใช้โครงสร้างข้อมูลแบบ record (ต่อ)

จงเขียนโปรแกรมรับตัวเลขรหัสของ 4 รายการข้างต้นของพนักงานแต่ละคน เพื่อคำนวณเงินเดือน จำนวน 120 คน

จำนวนปีที่ทำงาน	%	ความเอาใจใส่งาน	%
0 - 10	5	0 (ไม่ดี)	0
แต่ละปีถัดไป	4	1 (ดี)	10
		2 (ดีมาก)	25

Ex 2 จงใช้โครงสร้างข้อมูลแบบ record

เงินเดือนของพนักงาน XYZ ขึ้นอยู่กับ ฐานเงินเดือน ที่เท่ากันทุกคน คือ 3000 บาท บวกกับเปอร์เซ็นต์ของ ฐานเงินเดือนตาม ชนิดของงาน จำนวนปีที่ทำงาน ระดับความรู้ ความเอาใจใส่งาน

รายได้ (ic) = เงินเดือน + ฐานเงินเดือน * (% ชนิดงาน)

+ ฐานเงินเดือน * (%จำนวนปีที่ทำงาน)

+ ฐานเงินเดือน * (%ความเอาใจใส่งาน)



แบบฝึกหัด 2

BankAcc.pas

จงเขียน โปรแกรมต้องการนำทะเบียนลูกค้า(customer)
ของสหกรณ์ออมทรัพย์รามคำแหง ที่รองรับการจัดเก็บข้อมูล
ไม่เกิน 500 คน ทะเบียนลูกค้าประกอบด้วย รหัสบัญชี(id)
ชื่อบัญชี(name) ที่อยู่(Address) และเงินฝาก
(Deposit)



Menu

1. Open a new Account
2. Deposit
3. Withdraw
4. Compound interest
5. Quit

Please select 1 / 2 / 3 / 4 / 5 =?



เลือก 1 เป็นการเปิดบัญชีใหม่

ระบบจะแสดงหน้าจอให้ผู้ใส่ ป้อนรหัสบัญชี(id)

ชื่อบัญชี(name) ที่อยู่(Address) และเงินฝาก

(Deposit) โดย **รหัสบัญชีต้องไม่ซ้ำกัน**

เลือก 2 เป็นการฝากเงิน

โดยผู้ใช้ป้อนรหัสบัญชี(id)ที่ต้องการ **ระบบจะตรวจสอบ**

โดยการค้นหากระเป๋าสตางค์ เพื่อให้ผู้ใช้งานป้อนยอดฝาก

เพื่อปรับปรุงบัญชีให้ถูกต้อง

เลือก 3 เป็นการถอนเงิน

โดยผู้ใช้ป้อนรหัสบัญชี(id)ที่ต้องการ ระบบจะตรวจสอบเพื่อ

ค้นหากระเป๋เงินลูกค้า โดยการถอนเงินลูกต้องมีเงินติด

บัญชีอย่างน้อย 100 บาท ระบบทำการปรับปรุงบัญชีให้ถูกต้อง

เลือก 4 การคิดดอกเบี้ยทบต้น

1. โดยระบบจะให้ผู้ใช้ป้อนอัตราดอกเบี้ยต่อปี(rate) ทาง

แป้นพิมพ์ เพื่อคิดดอกเบี้ยทบต้นของทุกบัญชี

2. โปรแกรมจะแสดงเงินฝากเก่า , ดอกเบี้ยที่ได้รับ และเงินฝากใหม่

เงินฝากใหม่ = เงินฝากเดิม + ดอกเบี้ยที่ได้รับ

ดอกเบี้ยที่ได้รับ = อัตราดอกเบี้ยที่ได้รับ * เงินฝากเดิม

เลือก 5 เป็นการจบการทำงานซ้ำ

main.pas

```
1 program BankAccount;
2
3 type
4   Account = record
5     id: string[10];
6     name: string[100];
7     addr: string[100];
8     deposit: real;
9   end;
10
11 var
12   cust: array[1..500] of Account;
13   n, choice, i: integer;
14   id: string[10];
15   bfound: boolean;
16   deposit: real;
17
18 begin
19   n := 0;
20   repeat
21     writeln;
22     writeln('1. Open Acc');
23     writeln('2. Deposit Acc');
24     writeln('3. Withdraw Acc');
25     writeln('4. Compute Interest');
26     writeln('5. Exit');
27     write('Enter choice: ');
28     readln(choice);
29
```

BnkAcc.pas

```
30
31 1:
32 begin
33   writeln;
34   writeln('Do Open Acc');
35   write('id Acc =? ');
36   readln(id);
37   bfound := False;
38
39   for i := 1 to n do
40   begin
41     if cust[i].id = id then
42     begin
43       bfound := True;
44       break;
45     end; // end if
46   end; //end for
47
48   if bfound = False then
49   begin
50     n := n + 1;
51     cust[n].id := id;
52     write('name Acc =? ');
53     readln(cust[n].name);
54     write('addr Acc =? ');
55     readln(cust[n].addr);
56     write('deposit Acc =? ');
57     readln(cust[n].deposit);
58   end
59   else
60     writeln('id Acc found = ', id);
61
62 end; //case 1
63
```

```

64 2:
65   begin
66     writeln;
67     writeln('Do Deposit Acc');
68     write('id Acc =? ');
69     readln(id);
70
71     bfound := False;
72     for i := 1 to n do
73     begin
74       if cust[i].id = id then
75       begin
76         bfound := True;
77         break;
78       end; //end if
79     end; // end for
80
81     if bfound = False then
82       writeln('id Acc not found = ', id)
83     else
84     begin
85       writeln('Before deposit Acc = ', cust[i].deposit:0:2);
86       write('New deposit Acc =? ');
87       readln(deposit);
88       cust[i].deposit := cust[i].deposit + deposit;
89       writeln('After deposit Acc = ', cust[i].deposit:0:2);
90     end;
91   end; //case 2

```

```

92
93 3:
94   begin
95     writeln('Do Withdraw Acc');
96   end;
97
98 4:
99   begin
100     writeln('Do Interest Acc');
101   end;
102 end;
103
104 until choice = 5;
105 end.
106

```

BnkAcc.pas



□ END

การจัดเรียงข้อมูลในหน่วยความจำเมื่อมีการสร้างตัวแปร structure

1	2	1	2	3	4	1	2	3	4	1	2	3	4	5	6	7	8	9	10
a	g	t	e	l	l	w	e	i	g	n	a	m	e						
	e																		

```
type
PERSON = record
  age:integer;
  tell:integer;
  weight:real;
  name:string[10];
end;
```

Suppose int (2 bytes)
long ,float (4 bytes)
char (1) bytes