Departamento de Informática
Universidad de Valladolid
Campus de Segovia

POINTERS AND DINAMIC MEMORY ALLOCATION

P.1. Determine the output of the next code in Pascal:

```
    Program pointer;
    Type
        Pint: ^integer;
    Var
        I,J:Pint;
        N:integer;
    Begin
        New(I);
        New(j);
        N:=5;
        I^:=N;
        Writeln(I^);
        J:=I;
        J^:=-7;
        Writeln(I^)
    End.
```

P.2. Determine the output of the next code in Pascal:

```
Program test;
    Type
        student=record
            character:char;
            age:integer
        End;
        Ptest: ^student;
    Var
        P1,P2:Ptest;
    Begin
        New(P1);
        P1^.age:=1;
        P1^.character:= ‘A’
        Writeln(P1^.age,P1^.character);
        New(P2);
        P2^.Age:=1;
        P2^.Character:= ‘A’
```

```
                Writeln(P2^.Age,P2^.Character);
                P1:=P2;
                P2^.Age:=3;
                P2^.character:=’C’;
                Writeln(P1^.Age, P1^.character,P2^.Age,P2^.character)
End.
```

P.3. Determine the output of the next code in Pascal:

```
        Program simpletest;
        Var
                I,J:^integer
                x,y:integer;
        Begin
                New(I);
                New(j);
                I^:=5;
                x:=I^+1;
                y:=I^;
                New(J)
                J^:=y+3
                Writeln(x,’ ‘, y, ‘ ‘, I^, ‘ ‘, J^)
        End.
```

P.4. If “p” and “q” are pointers that point strings (^string) ¿What will be the output of netx code segment in Pascal?

```
                New(p);
                New(q);
                p^:=’Carpe’;
                q^=’Diem’;
                q:=p;
                q^:=’concha’;
                Writeln(p^);
                Writeln(q^);
```

P.5. Design and Implement a program in Pascal to work out the product between polynomials. The polynomial will be represented by means of vectors such as the zero-degree will be placed at the first vector position and so forth. The data structure and the headers of the procedures and functions are as follows:

```
        type
        tdimp=0..10;
        tdimpt=0..20;
        tArray=Array[tdimpt] of real;
        tApArray=^tArray;

        Procedure Polynomials_degree(var a,b:tAparray; var dima,dimb:tdimp);
        {it Asks the degrees of the two polynomials and initializes the vectors to zero}
        Procedure Enter_coeficient(var a,b:tApArray; dima,dimb:tdimp);
        {it Asks the coeficients of the two polynomials}
```

```
FUNCTION Polynomial_Product(a,b:tApArray;dima,dimb:tdimp):tApArray;
{it works out the product between two polynomials}
PROCEDURE Show_result(c:tApArray;dimt:tdimpt);
{it shows the result on the monitor screen}
```

P.6. Regarding the next data structure declaration:

```
TYPE
        Tname=string[20];
        Thobbies=(soccer,dancing,trekking,…);
        TsetHobbies=SET OF Thobbies;
        TpPerson=^Tperson;
        Tperson=Record
                                Name:Tname;
                                Hobbies:^TsetHobbies;
                End;
```

design and implement in Pascal the procedures and functions that will be neccessaries to store or delete the personal data (P:TpPerson) and taking into account a suitable memory management at any time. The headers of the procedures are as follows:

```
PROCEDURE Create_personal_data (VAR P:TpPerson);

PROCEDURE Delete_personal_data (VAR P:TpPerson);
```

Advise: The next implemented procedure can be used to load a set of hobbies into the variable “s”: read_set_of_Hobbies(VAR s:TsetHobbies).

P.7. Regarding the next data structure declaration:

```
CONST
        TopeTabla=20;
TYPE
        Tnombre=string[20];
        TpunNombre=^Tnombre;
        Tindice=1..TopeTabla;
        Ttabla=array[TIndice] of TpunNombre;
```

And taking into account that the table is filled up and every pointer is pointing a name, design and implement in Pascal a procedure to find the smallest name (regarding the alphabetic order) and change its actual position with the first one, if both of them are different. The header of the procedure is as follows:
:

```
PROCEDURE change_smallest_name ( VAR tabla:Ttabla)
```