SOLUTION FOR STRUCTURES AND UNION

1) Write a program to define structure with tag `state` with fields `state name`, number of districts and total population. Read and display the data.

```c
#include <stdio.h>
#include <conio.h>

struct state
{
    char *name;
    int dist;
    long pop;
};

void main()
{
    clrscr();
    state s;
    printf ("Enter state name, number of dist. and population : ");
    scanf ("%s %d %ld", s.name, &s.dist, &s.pop);
    printf ("State : %s", s.name);
    printf ("District : %d", s.dist);
    printf ("Population : %ld", s.pop);
}

OUTPUT
Enter state name, number of dist. and population : ms 20 700000
State : ms
District : 20
Population : 700000

Explanation In the above program `struct state` is defined with the members `name`, `dist`, and `pop`. Using `scanf()` statement data is read. The `printf()` statement displays the data on the screen. The dot operator is used to access the elements.

2) Write the program (1)-using pointer to structure.

```c
#include <stdio.h>
#include <conio.h>

struct state
{
    char *name;
    int dist;
    long pop;
};

void main()
{
    clrscr();
```
state *s;
printf ("Enter state name, number of dist. and population : ");
scanf ("%s %d %ld", s->name,&s->dist,&s->pop);
printf ("State : %s",s->name);
printf ("District : %d",s->dist);
printf ("Population : %ld",s->pop);
}

OUTPUT
Enter state name, number of dist. and population : ms 20 700000
State : ms
District : 20
Population : 700000

Explanation
The above program is same as last one. Here, object pointer *s is declared. The arrow (->) operator is used to access the elements.

3) Define a structure with tag population with fields Men and Women. Create structure with in structure using state and population structure. Read and display the data.

#include <stdio.h>
#include <conio.h>

struct state
{
    char *name;
    int dist;
    long pop;
};

struct population
{
    long men;
    long women;
    struct state s1;
};

void main()
{
    struct population p;
    clrscr();
    printf ("State : "); scanf ("%s",p.s1.name);
    printf ("Districts : "); scanf ("%d",&p.s1.dist);
    printf ("Men : "); scanf ("%ld",&p.men);
    printf ("Women : "); scanf ("%ld",&p.women);
    p.s1.pop=p.men+p.women;
    printf ("State : %s ",p.s1.name);
    printf ("Districts : %d",p.s1.dist);
    printf ("Men : %ld",p.men);
# include <stdio.h>
# include <conio.h>
struct state
{
    char *name;
    int dist;
    long pop;
};
struct population
{
    long men;
    long women;
    struct state s1;
};
void main()
{
    struct population p[2];
    int i;
clearscr();
for (i=0;i<2;i++)
{ printf ("State : %s ",p.s1.name);
    printf ("Men : %ld ",p.men);
    printf ("Women : %ld ",p.women);
    printf ("Population :%ld",p.s1.pop);
}
{ printf ("State : "); scanf ("%s",p[i].s1.name);  
printf ("Districts : "); scanf ("%d",&p[i].s1.dist); 
printf ("Men : "); scanf ("%ld",&p[i].men);  
printf ("Women : "); scanf ("%ld",&p[i].women); 
p[i].s1.pop=p[i].men+p[i].women; 
} for (i=0;i<2;i++)
{
printf ("State : %s ",p[i].s1.name); 
printf ("Districts : %d",p[i].s1.dist); 
printf ("Men : %ld",p[i].men);  
printf ("Women : %ld",p[i].women); 
printf ("Population :%ld\n",p[i].s1.pop); 
}

OUTPUT
State : MS
Districts : 25
Men : 500000
Women : 400000
State : AP
Districts : 20
Men : 450000
Women : 300000
State : MS
Districts : 25
Men : 500000
Women : 400000
Population :900000
State : AP
Districts : 20
Men : 450000
Women : 300000
Population :750000

Explanation The above program is same as last one. In this program, an array of two elements is declared. Using for loops repetitive read and display operations are performed.

5) Create user-defined data type equivalent to int. Declare 3 variables of its type. Perform arithmetic operations using these variables.

#include <stdio.h>
#include <conio.h>
void main()
{

typedef int num;
num a,b,c;
clrscr();
printf ("\n Enter three Numbers : ");
scanf ("%d %d %d", &a,&b,&c);
printf ("\n Addition : %d",a+b+c);
printf ("\n Subtraction : %d",a-b-c);
printf ("\n Multiplication : %d", a*b*c);
}

OUTPUT
Enter three Numbers: 2 3 4
Addition : 9
Subtraction : -5
Multiplication : 24

Explanation: Using typedef statement, a user defined data type num equivalent to basic type int is declared. The variables a, b and c are of num type. Using scanf (0 statement three numbers are entered. The printf () statement displays addition, multiplication, and subtraction of these numbers.

6) Create enumerated data type logical with TRUE and FALSE values. Write a program to check whether the entered number is positive or negative. If the number is positive, display one otherwise zero. Use enumerated data type logical to display 0 and 1.

#include <stdio.h>
#include <conio.h>
void main()
{
enum logical { false,true};
int n;
clrscr();
printf ("\n Enter a Number : ");
scanf ("%d",&n);
if (n>0)
printf ("%d",true);
else
printf ("%d",false);
}

OUTPUT
Enter a Number : -1
  0

Explanation: An enumerated data type logical with two values (true (1) and false (0)) is declared. The entered number is checked. If it is positive, one is displayed otherwise; zero is displayed using enum data type.
7) Write a program to accept records of the different states using array of structures. The structure should contain char state, int population, int literacy rate, and int per capita income. Assume suitable data. Display the state whose literacy rate is highest and whose per capita income is highest.

```c
#include <stdio.h>
#include <conio.h>
struct data
{
    char state[10];
    long pop;
    int lit_rate;
    int capita;
};
void main()
{
    struct data r[3];
    int i,hl=0,hi=0,x,y;
    clrscr();
    for (i=0;i<=2;i++)
    {
        printf ("State : "); scanf ("%s",r[i].state);
        printf (" Population : "); scanf ("%ld",&r[i].pop);
        printf (" Literacy rate : "); scanf ("%d",&r[i].lit_rate);
        printf (" Per capita Income : "); scanf ("%d",&r[i].capita);
        if (i>0)
        {
            if (r[i].lit_rate>hl)
            {
                hl=r[i].lit_rate;
                x=i;
            }
            if (r[i].capita>hi)
            {
                hi=r[i].capita;
                y=i;
            }
        }
        else
        {
            hl=r[i].lit_rate;
            hi=r[i].capita;
        }
    }
    printf (" State of Highest Literacy Rate ");
```
printf ("\nState : %s",r[x].state);
printf ("\nPopulation : %ld",r[x].pop);
printf ("\nLiteracy rate : %d",r[x].lit_rate);
printf ("\nPer capita Income : %d",r[x].capita);
printf ("\n State of Highest per capita income ");
printf ("\nState : %s",r[y].state);
printf ("\nPopulation : %ld",r[y].pop);
printf ("\nLiteracy rate : %d",r[y].lit_rate);
printf ("\nPer capita Income : %d",r[y].capita);
}

OUTPUT
State : MS
Population : 700000
Literacy rate : 55
Per capita Income : 2000
State : AP
Population : 650000
Literacy rate : 49
Per capita Income : 1500
State : UP
Population : 600000
Literacy rate : 58
Per capita Income : 2500
State of Highest Literacy Rate
State : UP
Population : 600000
Literacy rate : 58
Per capita Income : 2500
State of Highest per capita income
State : UP
Population : 600000
Literacy rate : 58
Per capita Income : 2500

Explanation In this program struct data is declared. The members are as per stated in exercise. An array of structure variable r [3] is declared. Information of three states is entered. During information reading, the literacy rate and per capita income of states are compared with one another using if () statement ladder inside the loop. The variable x and y are used to store array element number of records, which contain highest rate. Using printf () statement both records are displayed.

8) Write a program to accept records of different states using array of structures. The structure should contain char State and number of int engineering collages, int medical collages, int management collages and int universities. Calculate total collages and display the state, which is having highest number of collages.
# include <stdio.h>
# include <conio.h>

struct data
{
    char state[10];
    int eng_col;
    int med_col;
    int mamg_col;
    int uni;
    int total;
};

void main()
{
    struct data r[3];
    int i, hc=0, x=0;
    clrscr();
    for (i=0; i<2; i++)
    {
        printf ("State : "); scanf ("%s", r[i].state);
        printf (" Engineering Colleges : "); scanf ("%d", &r[i].eng_col);
        printf (" Medical Colleges : "); scanf ("%d", &r[i].med_col);
        printf (" Management Colleges : "); scanf ("%d", &r[i].mamg_col);
        printf (" Univrsities : "); scanf ("%d", &r[i].uni);
        r[i].total=r[i].eng_col+r[i].med_col+r[i].mamg_col+r[i].uni;
        printf ("Total : %d", r[i].total);
        if (i>0)
        {
            if (r[i].total>hc)
            {
                hc=r[i].total;
                x=i;
            }
        }
        else
        {
            hc=r[i].total;
            x=i;
        }
    }
    printf ("State having maximum colleges 
State : ");
    printf ("Engineering Colleges : %d", r[x].eng_col);
    printf ("Medical Colleges : %d", r[x].med_col);
    printf ("Management Colleges : %d", r[x].mamg_col);
    printf ("Univrsities : %d", r[x].uni);
printf("\n Total colleges : %d",r[x].total);
}

OUTPUT
State : MS
Engineering Colleges : 200
Medical Colleges : 100
Management Colleges : 150
Universities : 20
Total : 470
State : AP
Engineering Colleges : 150
Medical Colleges : 175
Management Colleges : 125
Universities: 15
Total : 465
State having maximum colleges
State : MS
Engineering Colleges : 200
Medical Colleges : 100
Management Colleges : 150
Universities : 20
Total colleges : 470

Explanation The logic of the above program is same as last one. Here, only one value is compared out of two records.

9) Write a program to check status of the printer using int86() function. The details are as per given below.
   a) Interrupt - 0x17
   b) Inputs - AH = 0x02
      - DX = printer port number (0=LPT1, 1=LPT2, 2=LPT3)
   c) Returns - AH = Completion / nonsuccess code

   Completion values are
   Bit 0=1 : time out
   Bit 3=1 : I/O mistake
   Bit 4=1 : Printer selected
   Bit 5=1 : Out of paper
   Bit 6=1 : Printer acknowledge
   Bit 7=1 : Printer not engaged
   # include <stdio.h>
   # include <conio.h>
   # include <dos.h>
   void main ()
   {
   union REGS in,out;
   clrscr();
in.h.ah=2;
in.x.dx=0;
int86(0x17,&in,&out);
out.h.ah &=0x80;
if (out.h.ah==0) printf ("\n Printer is not ready");
else printf ("\n Printer is ready");
}

OUTPUT

Printer is not ready

Explanation In the above program given values are placed in the CPU registers and using int86() function interrupt is invoked. If the value of ah register is zero, then the printer is ready and user can perform the printing operation. If ah contains not-zero value it means that the printer is not ready.

10) Write a program to reboot the system. Use following data with int86() function.
   a) Interrupt - 0x19
   b) Input - Void (Nothing)

# include <stdio.h>
# include <conio.h>
# include <dos.h>
void main ()
{
    union REGS in,out;
    clrscr();
    int86(0x19,&in,&out);
}

Explanation In the above program the interrupt 0x19 is invoked. The system ois rebooted. Some time only the current program is closed.