

1(c) Nested structure means structure within a structure. Say we create a structure for student and the members include name, roll no , 3 subject marks and total. But name in turn can be a structure of 3 members: first name, middle name and last name. The structure can be specified as follows

```
struct details
{
    char fname[20];
    char mname[20];
    char sname[20];
};
struct student
{
    struct details name;
    int id, m1,m2,m3,t;
};
```

Alternately it can also be done as

```
struct student
{
    struct
    {
        char fname[20];
        char mname[20];
        char sname[20];
    }name;
    int id, m1,m2,m3,t;
};
```

here is a program that reads the details of a student and prints it.

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

```
struct details
{
    char fname[20];
    char mname[20];
    char sname[20];
};
struct student
{
    struct details name;
    int id, m1,m2,m3,t;
};
```

```
void main()
{
    struct student x;

    printf("Please enter first name");
    scanf("%s",x.name.fname);
    printf("Please enter middle name");
    scanf("%s",x.name.mname);
    printf("Please enter surname");
    scanf("%s",x.name.sname);

    printf("Please enter id, 3 subject marks");
    scanf("%d%d%d%d",&x.id,&x.m1,&x.m2,&x.m3);

    x.t=x.m1+x.m2+x.m3;
    printf("Name: %s %s %s\nId=%d\nSub1=%d\nSub2=%d\nSub3=%d\nTotal=%d\n",
           x.name.fname,x.name.mname,x.name.sname,x.id,x.m1,x.m2,x.m3,x.t);
}
```

**Navlakhi®**

1(e) goto: This is used to transfer control from one place in the program to another in the same program unconditionally

```
#include<stdio.h>
void main()
{
    printf("This is line 1\n");
    goto a;
    printf("This is line 2\n");
a: printf("This is line 3\n");
}
```

Output:

This is line 1

This is line 3

continue: This is used to go to the next iteration of the loop before completing the body of the loop

```
#include<stdio.h>
void main()
{
    int i;
    for(i=1;i<=10;i++)
    {
        if (i%2==0)
            continue;
        else
            printf("%d\n",i);
    }
}
```

OUTPUT:

1  
3  
5  
7  
9

2(a) count number is vowels using switch case

```
#include<stdio.h>
#include<string.h>
void main()
{
    int c=0,i;
    char x[100];

    printf("enter a word ");
    scanf("%s",x);

    l=strlen(x);

    for(i=0;i<l;i++)
    {
        switch(x[i])
        {
            case 'A':
            case 'a':
            case 'E':
            case 'e':
            case 'I':
            case 'i':
            case 'O':
            case 'o':
            case 'U':
            case 'u':
                c=c+1;
                break;
        }
    }
    printf("Number of vowels=%d\n",c);
}
```

3a(ii)

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

```
void main()
```

```
{
```

```
    int n, x[100], digit,i,k;
```

```
    printf("enter a number ");
```

```
    scanf("%d",&n);
```

```
    i=0;
```

```
    while(n!=0)
```

```
    {
```

```
        digit=n%2;
```

```
        n=n/2;
```

```
        x[i]=digit;
```

```
        i=i+1;
```

```
    }
```

```
    for(k=i-1;k>=0;k--) /*printing in reverse*/
```

```
    {
```

```
        printf("%d",x[k]);
```

```
    }
```

```
}
```



3(b)

```
#include<stdio.h>
int symm(int x[100][100],int m, int n)
{
    int success=1,i,j;

    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            if(x[i][j]!=x[j][i])
                success=0;
        }
    }
    return success;
}

void main()
{
    int m,n, x[100][100], i,j,ans;

    printf("enter number of rows and cols ");
    scanf("%d%d",&m,&n);

    if(m!=n)
    {
        printf("Not a square matrix");
    }
    else
    {
        printf("Enter the matrix\n");
        for(i=0;i<m;i++)
        {
            for(j=0;j<n;j++)
            {
                scanf("%d",&x[i][j]);
            }
        }
        ans=symm(x,m,n);

        if(ans==1)
            printf("It is a symmetric matrix");
        else
            printf("It is not a symmetric matrix");
    }
}
```



4(A)

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

```
#include<math.h>
```

```
void main()
```

```
{
```

```
    int n,i;
```

```
    float x[100], sum=0, sum2=0,meanx, sd;
```

```
    printf("enter number of elements ");
```

```
    scanf("%d",&n);
```

```
    printf("Enter the elements\n");
```

```
    for(i=0;i<n;i++)
```

```
    {
```

```
        scanf("%f",&x[i]);
```

```
        sum=sum+x[i];
```

```
    }
```

```
    meanx=sum/n;
```

```
    for(i=0;i<n;i++)
```

```
    {
```

```
        sum2=sum2+(x[i]-meanx)*(x[i]-meanx);
```

```
        printf("%f\t%f\n",x[i],sum2);
```

```
    }
```

```
    sd=sqrt(sum2/n);
```

```
    printf("Standard deviation=%f\n",sd);
```

```
}
```

4(b)(i)

(1) strncpy(destination array, source array, length to copy);

example

```
#include <stdio.h>
#include <string.h>
void main()
{
    char y[10],x[10];
    strcpy(x,"program");
    strncpy(y,x, 4); /*copies 4 characters of x into y*/
    y[4] = '\0';    /*y[0] to y[3] is filled so NULLing y[4] */
    printf("%s\n",y);
}
```

Output:

prog

(2) strcat .....

```
(ii)
#include<stdio.h>
#include<string.h>
void main()
{
    char x[100],y[100];
    int l,i,j;

    printf("Enter word ");
    scanf("%s",x);

    l=0;
    while(x[l]!='\0')
    {
        l=l+1;
    }

    for(i=0,j=l-1;i<=l-1;i++,j--)
    {
        y[j]=x[i];
    }
    y[l]='\0';

    printf("Reverse=%s\n",y);
}
```

5(a)

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

```
#include<string.h>
```

```
struct player
```

```
{
```

```
    char name[20];
```

```
    char country[20];
```

```
    float avg;
```

```
};
```

```
void main()
```

```
{
```

```
    struct player x[20];
```

```
    int i;
```

```
    char s[20];
```

```
    for(i=0;i<20;i++)
```

```
    {
```

```
        printf("Enter player name ");
```

```
        scanf("%s",x[i].name);
```

```
        printf("Enter country name ");
```

```
        scanf("%s",x[i].country);
```

```
        printf("Enter average ");
```

```
        scanf("%f",&x[i].avg);
```

```
    }
```

```
    printf("Enter the name of player to search for ");
```

```
    scanf("%s",s);
```

```
    for(i=0;i<20;i++)
```

```
    {
```

```
        if(strcmp(x[i].name,s)==0)
```

```
        {
```

```
            printf("Name=%s\nCountry=%s\nAverage=%f\n",x[i].name, x[i].country, x[i].avg);
```

```
        }
```

```
    }
```

```
}
```



5(b)

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

```
int recsum(int term, int n)
```

```
{  
    if(term==n)  
        return term;  
    else  
        return term+recsum(term+1,n);  
}
```

```
void main()
```

```
{  
    int n,ans;  
    printf("Enter n ");  
    scanf("%d",&n);  
  
    ans=recsum(1,n);  
  
    printf("Sum=%d\n",ans);  
}
```



6(b)

1. Start
2. Accept n
3.  $i:=2$
4. Display 1 (since 1 is a factor of all numbers)
5. while( $n \neq 1$ )
  - 5.1 if  $n \text{ MOD } i = 0$  then
    - 5.1.1 display i
    - 5.1.2  $n:=n/i$
  - 5.2 else
    - 5.2.1  $i:=i+1$
6. STOP



**Now Access to all our websites with  
superfast hosting**

**Launching soon**

**Navlakhi.fashion &  
Navlakhi.studio**