

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#define pi 3.14
```

```
void cylinder(float *, float *);
void cone(float *, float *);
void sphere(float *, float *);
```

```
int main()
```

```
{
```

```
float area, volume;
```

```
int choice, flag;
```

```
do
```

```
{
```

```
printf("\n Enter the choice of shape in it
```

```
1 → cylinder It 2 → Cone It 3 → sphere\n choice:");
```

```
scanf("%d", &choice); switch(choice)
```

```
{
```

```
case 1: cylinder(&area, &volume);
break;
```

```
case 2: cone(&area, &volume);
break;
```

```
case 3: sphere(&area, &volume);
break;
```

```
default: printf("\n Input Error!! ");
exit(0);
```

```
}
```

```
printf("\n The Area = %.2f And Volume = %.2f\n",
area, volume);
```

```
printf("\n Press 1 to continue: ");
```

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

```
} while (flag == 1);
```

```
return 0;
```

```
}
```

```
void cylinder (float *a, float *v)
{
```

```
    float r, h;
    printf ("Enter The Radius and Height: \n");
    scanf ("%f %f", &r, &h);
    *a = (2 * pi * r) * (r + h);
    *v = pi * r * r * h;
```

```
}
```

```
void cone (float *a, float *v)
```

```
{
    float r, h;
    printf ("Enter The Radius and Height: \n");
    scanf ("%f %f", &r, &h);
    *a = (pi * r) * (r + sqrt(r * r + h * h));
    *v = pi * r * r * h / 3;
```

```
}
```

```
void sphere (float *a, float *v)
```

```
{
    float r;
    printf ("Enter The radius of sphere, in radius: \n");
    scanf ("%f", &r);
    *a = 4 * pi * r * r;
    *v = pi * r * r * r * 4 / 3;
```

```
}
```

7.

```
#include <stdio.h>

struct student
{
    char Name[30];
    int Elective;
};

void getdata (struct student *, int);
int count (struct student *, int, int);
void check (int *, struct student *, int);
void se-getdata (struct student *, int, int *, int);
void display (struct student *, int *, int);

int main()
{
    int n, E[3];
    printf("Enter the No. of students : ");
    scanf("%d", &n);
    struct student S[n];
    getdata(S, n);

    for (int i = 1; i <= n; i++)
    {
        E[i-1] = count(S, i, n);
    }
    check(E, S, n);
    display(S, E, n);

    return 0;
}

void getdata (struct student * S, int n)
{
    for (int i = 0; i < n; i++)
    {
        printf("Enter the Name of student &
            choice of Elective : ");
        scanf("%s %d", &S[i].Name, &S[i].Elective);
    }
}
```

```

int count (struct student *S, int k, int n)
{
    int c=0;
    for (int i=0; i<n; i++)
        if (S[i].Elective == k)
            c++;
    return c;
}

```

```

void check (int *E, struct student *S, int n)
{
    printf("\n The current Electives are: IOT: %d\n", E[0]);
    printf("Advanced Java & J2EE: %d\n", E[1]);
    printf("Advanced Data Structure: %d\n", E[2]);

    int net;
    net = (int) (.3 * n);

    for (int i=1; i<=3; i++)
    {
        if (E[i-1] < net)
        {
            printf("\n Elective no %d is not available\n", i);
            printf("recheck the Elective no %d\n", i);

            re-getdata (S, i, E, n);
            if (E[i-1] == 0)
            {
                printf("\n Elective no %d is no more\n", i);
            }
        }
    }
}

```

```

void re-getdata (struct student *S, int k, int *E, int n)
{
    int temp;
    for (int i=0; i<n; i++)
        if (S[i].Elective == k)
        {
            printf("\n %s: Enter other than %d elective: ", S[i].Name, k);

            scanf ("%d", &temp);
            S[i].Elective = temp;
            E[temp-1]++; E[k-1]--;
        }
}

```

}

```
void display (struct student *s, int *E, int n)
{
```

```
    for (int i=1; i<=3; i++)
```

```
    { printf ("In the No of students in (%d) elective are : %d\n", i, E[i-1]);
```

```
        for (int j=0; j<n; j++)
```

```
            if (S[j].elective == i)
```

```
                printf ("< %s> \n", S[j].Name);
```

```
    }
```

```
}
```