(2) Variable space requirements: Space needed by structured variables whose size depends on the particular instance, I, of the problem being solved.

- Also includes the additional space reqd. when a function uses recursion
- $S_p(I)$: Variable space requirement of a program P working on an instance I.
- Usually given as a function of some characteristics of the instance I. Ex: no.; size, values of the inputs and outputs assosciated with I.
- We can express the total space requirement S(P) of any program as: $\boxed{S(P) = c + S_p(I)}$ where c is a constant representing the fixed space requirements. We are usually concerned with variable space requirements.

Ex 1.6:

```
float abc(float a, float b, float c)
{
    return a+b+b*c+(a+b-c)/(a+b)+4.00;
}
```

According to the classification given, this function has only fixed space requirements. $\boxed{\therefore S_{abc}(I) = 0}$

Ex 1.7:

```
float sum(float list[], int n)
{
    float tempsum=0;
    int i;
    for(i=0; i<n; i++)
        tempsum+= list[i];
    return tempsum;
}
```

- The variable space requirement depends on how the array is passed into the function.
- Programming languages like Pascal may pass arrays by value. This means that the entire array is copied into temporary storage before the function is executed. In these languages,

$S_{sum}(I) = S_{sum}(n) = n$, where n is the size of the array.

- When an array is passed as an argument to a function, C interprets it as passing the address of the first element of the array. C does not copy the array. $\therefore S_{sum}(n) = 0$.