```
/*
Εξηγήστε τι κάνει η ακόλουθη συνάρτηση και δώστε μία υλοποίηση
χωρίς αναδρομή:

    int test(const int b[], int  p)
    {
    if (p==1) {
        return b[0];
       }
    else {
         return b[p-1]+test(b,p-1);
         }
     }

Φτιάξτε κυρίως πρόγραμμα που γεμίζει τον πίνακα με θετικά
στοιχεία (αμυντικός προγραμματισμός) και καλεί την υλοποιημένη
από εσάς συνάρτηση.

*/

#include <stdio.h>
#define N 5

int test1(const int b[], int  p)
    {
    if (p==1) {
        return b[0];
       } /* end if */
    else { /* recursion step */
         return b[p-1]+test1(b,p-1);
         }
    }

int test2(const int b[], int  p)
{

int i;
int sum=0;

for (i=0; i<p; i++)
    sum=sum+b[i];

return sum;

}


int main()
{
    int data[N];
    int element;
    int thesi;
```

```
    int i;

    for (i=0; i<N; i++)
    {
        do
        {
            printf("Give %d number (>=0: ", i);
            scanf("%d", &data[i]);
        }
        while (data[i]<0);

    }

int sum=0;

    sum=test1(data, N);
    printf("The sum is %d\n", sum);

    sum=test2(data, N);
    printf("The sum is %d\n", sum);


    return 0;
}
```