

```

1  #include <iostream>
2  #include <algorithm>
3  #include <vector>
4
5  /*Dado un vector V de n enteros distintos, y un n° S, determinar si existen y
6  cuales son 2 numeros en el vector que sumados, den el valor S
7  solución: 1º se ordena por quicksort y despues se aplica BusBinaria.
8  Evidentemente nos tendrían que facilitar el quicksort por cuestion de tiempo!!
9  o llamamos a sort de C++ q hace lo mismo.
10 Y el BusqBinaria podríamos hacerlo "nosotros". :)
11 */
12
13 #define umbral 2 //umbral de Quicksort
14
15 using namespace std;
16
17
18
19
20
21 int BusquedaBinariaRekursiva(int x,vector<int> v,int ini,int fin){
22
23     int mitad;
24     if(ini>fin){
25         return (-1);
26     }else{
27         if(ini==fin){
28             if(x==v[ini]){
29                 return (ini);
30             }else{
31                 return (-1);
32             }
33         }else{
34
35             mitad=(ini+fin)/2;
36             if(x==v[mitad]){
37                 return(mitad);
38             }else{
39                 if(x<v[mitad]){
40                     return (BusquedaBinariaRekursiva(x,v,ini,mitad-1));
41                 }else{
42                     return(BusquedaBinariaRekursiva(x,v,mitad+1,fin));
43                 }
44             }
45         }
46     }
47 }
48
49
50 //Este módulo no es recursivo pero es una aplicación
51 //muy buena del busqueda binaria. ;)
52 bool BuscaSuma2(vector<int> v, int S, int &x1, int &x2){
53     int res;
54
55     sort(v.begin() ,v.end()); //ordenamos vector
56     int j=0;
57     bool existe=false;
58     while (j<v.size() && !existe){
59         x1=v[j];
60         x2=S-x1; //buscamos x2 en v, que es el que nos falta para cumplir objetivo
61
62         res=BusquedaBinariaRekursiva(x2,v,0,v.size()-1 );
63
64         if (res!=-1 && res!=j) //si lo encuentra y no es el mismo repetido
65             existe=true;
66         j++;

```

```
67     }
68     return existe;
69 }
70
71 int main()
72 {
73     int S=15;
74     int x1,x2;
75     vector<int> v;
76     v.push_back(1);
77     v.push_back(4);
78     v.push_back(3);
79     v.push_back(5);
80     v.push_back(12);
81     v.push_back(6);
82
83     cout << "valor buscado: " << S << endl;
84     if (BuscaSuma2(v,S,x1,x2))
85         cout << "Numero encontrados:" << x1 << ", " << x2 << endl;
86     else
87         cout << "No existen dichos numeros" << endl;
88     return 0;
89 }
```