

1)

(1)
3.2 3

(2)
3.2 3
4.1 4

(3)
3.2 3
4.1 4
3.8 3

(4)
2 14

(5)
3.2 3
4.1 4
3.8 3
11 7

2)

(1)
A
A
B
C
D
A
B

(2)
2 0
D 3 2

(3)
2 0
A 0

(4)
1 0
B 1

(5)
1 0
A 0

3) 110101011010
10010110

1111101001001010

4)

```
bool ordinato(int v[], int n)
{
    if (n == 1) return true;
    if (v[n-1] < v[n-2]) return false;
    return ordinato(v, n-1);
}
```

5)

```
void elimina (elem*& p0, int x)
{   elem *p = p0,*q;
    while (p != 0)
    {   if (p->info == x)
        {   if (p0 == p)
            {   p0 = p->pun;
                delete p;
                p = p0;
            }
            else
            {   q->pun = p->pun;
                delete p;
                p = q->pun;
            }
        }
        else
        {   q = p;
            p = p->pun;
        }
    }
}
```

6)

```
void inserisci(int v[], int n, int x)
{
    int i,j;
    for (i = 0; i < n && v[i] <= x; i++);
        // x maggiore di tutti gli elementi
    if (i == n) return;
        // shift elementi a destra di una posizione
    for (j = n-1; j > i; j--)
        v[j] = v[j-1];
    v[i] = x;
}
```