

Esercizio 1)

```

int maggiore (int el, int a[], int inizio, int fine)
{
    int i = inizio;
    int trovato = 0;
    while (i < fine && !trovato)
        if (a[i] > el) trovato = 1;
        else i++;
    return trovato;
}

```

```

int check (int a[], int dim)
{
    int i = 1;
    int trovato = 0;
    while (i < dim - 1 && !trovato)
        if (maggiore (a[i], a, 0, i) &&
            maggiore (a[i], a, i + 1, dim)) trovato = 1;
        else i++;
    return trovato;
}

```

```

int maggiore (int el, int a[], int inizio, int fine)
{
    int i = inizio;
    int trovato = 0;
    while (i < fine && !trovato)
        if (a[i] > el) trovato = 1;
        else i++;
    return trovato;
}

```

```

void num2 (int a[], int dim, int *num)
{
    int i;
    *num = 0;
    for (i = 1; i < dim - 1, i++)
        if (maggiore (a[i], a, 0, i) &&
            maggiore (a[i], a, i + 1, dim)) (*num)++;
}

```

```
void swap(int *x, int *y)
{
    int temp = *x;
    *x = *y;
    *y = temp;
}
```

```
void move(int a[], int dima)
{
    int i;
    int d = dima % 2;
    for (i = 0; i < dima/2; i++)
        swap(&a[i], &a[i+dima/2+d]);
}
```

d ha il valore 0 se *dima* è pari e il valore 1 se *dima* è dispari.