

Znenie zadania:

Implementujte algoritmus BubbleSort. Postupne zobrazujte činnosť algoritmu (stav poľa) po každom prechode hlavným cyklom. Pridajte podporu utriedenia vstupnej postupnosti vzostupne a zostupne podľa hodnoty parametra funkcie realizujúcej triedenie

Opis riešenia:

Algoritmus BubbleSort porovnáva postupne každú dvojicu prvkov. Túto dvojicu porovnáva a dáva do správneho poradia. Takéto porovnania sa vykonajú toľkokrát, koľko je prvkov na zotriedenie.

Zdrojový kód:

funkcia bubbleSort(int a[], int n, int type):

```
void bubbleSort(int a[], int n, int type){
    int i, j, k;

    if (type == 0){
        printf("bubbleSort - vzostupne\n");
        for (i = 1; i<=n; i++){
            for (j=n; j>=i+1; j--){
                if (a[j-1]>a[j]) {
                    a[j] -= a[j-1];
                    a[j-1] += a[j];
                    a[j] = a[j-1]-a[j];
                }
            }
            for(k=1;k<=n;k++)printf("%2d ",a[k]);
            printf("\t( i = %d)\n",i);
        }
    }else{
        printf("bubbleSort - zostupne\n");
        for (i = 1; i<=n; i++){
            for (j=n; j>=i+1; j--){
                if (a[j-1]<a[j]) {
                    a[j] -= a[j-1];
                    a[j-1] += a[j];
                    a[j] = a[j-1]-a[j];
                }
            }
            for(k=1;k<=n;k++)printf("%2d ",a[k]);
            printf("\t( i = %d)\n",i);
        }
    }
}
```

Znenie zadania:

Implementujte algoritmus InsertionSort. Postupne zobrazujte činnosť algoritmu (stav poľa) po každom prechode hlavným cyklom. Pridajte podporu utriedenia vstupnej postupnosti vzostupne a zostupne podľa hodnoty parametra funkcie realizujúcej triedenie

Opis riešenia:

Algoritmus InsertSort prechádza postupne všetky prvky. Postupne vyberá jeden prvok ktorý porovnáva s ostatnými až kým nie je daný prvok utriedený. Takto sa prejde všetkými prvkami poľa.

Zdrojový kód:

funkcia insertSort (int a [], int n, int type):

```
void insertSort(int a[], int n, int type){
    int i, j, k, temp;
    if (type == 0){
        printf("insertSort - vzostupne\n");
        for (i = 1; i<=n; i++){
            temp = a[i];
            j = i-1;
            while(j >= 0 && a[j]>temp){
                a[j+1] = a[j];
                j--;
            }
            a[j+1] = temp;
            for(k=1;k<=n;k++)printf("%2d ",a[k]);
            printf("\t( i = %d)\n",i);
        }
    }else{
        printf("insertSort - zostupne\n");
        for (i = 1; i<=n; i++){
            temp = a[i];
            j = i-1;
            while(j >= 1 && a[j]<temp){
                a[j+1] = a[j];
                j--;
            }
            a[j+1] = temp;
            for(k=1;k<=n;k++)printf("%2d ",a[k]);
            printf("\t( i = %d)\n",i);
        }
    }
}
```