9.Cvičenie

BubbleSort

```
// Panik
void bubbleSort(int a[], int n, int dir)
{
	for(int i = 1; i <= n; i++) {
		for(int j = n; j > i; j--) {
			if(dir && a[j - 1] > a[j]) {
				int temp = a[j - 1];
				a[j - 1] = a[j];
				a[j] = temp;
			}
			else if(!dir && a[j - 1] < a[j]) {
				int temp = a[j - 1];
				a[j - 1] = a[j];
				a[j] = temp;
			}
		}
		PrintArray(a, n, dir);
	}
}

//Kolosovskyi
void bubbleSort(int a[], int n, int dir)
{
   if(a == NULL || (dir != 0 && dir != 1)) {
      return;
   }
   int buf;
   for(int i = 1; i <= n; i++){
      for(int j = n; j >= i + 1; j--) {
         if(dir == 1 ? a[j-1] > a[j] : a[j-1] < a[j]) {
            buf = a[j-1];
            a[j-1] = a[j];
            a[j] = buf;
         }
      }
      PrintArray(a, n, dir);
   }
}
```

```
//LENGYEL
void bubbleSort(int a[], int n, int dir)
{
   if (n < 2)
         return;

   for (int i = 1; i <= n; i++)
   {
         for (int j = n; j >= i + 1; j--)
         {
               if (dir == 0 ? a[j] > a[j - 1] : a[j] < a[j - 1])
               {
                     int temp = a[j - 1];
                     a[j - 1] = a[j];
                     a[j] = temp;
               }
         }
         PrintArray(a, n, 0);
   }
}
```

InsertSort

```
// Panik
void insertSort(int *A, int n, int dir)
{
         int i, j, value;
         for(i = 1; i <= n; i++) {
               value = A[i];
               j = i-1;
               if(dir){
                     while(j >= 1 && A[j] > value) {
                           A[j + 1] = A[j];
                           j = j - 1;
                     }
               }
               else {
                     while(j >= 1 && A[j] < value) {
```

```
                                A[j + 1] = A[j];
                                j = j - 1;
                            }
                        }

                        A[j+1] = value;
                        PrintArray(A, n, dir);
                }
}


// Maňkoš
void insertSort(int *A, int n, int dir) {
    if (A && (dir == ASC || dir == DSC)) {
        for (int i = 1; i <= n; i++) {
            int val = A[i];
            int j = i - 1;
            for (; j > 0 && ((dir == ASC && A[j] > val) || (dir == DSC && A[j] < val)); A[j+1] = A[j], j--)
{};
            A[j+1] = val;
            PrintArray(A, n, i);
        }
    }
}

//Kolosovskyi
void insertSort(int *A, int n, int dir)
{
    if(A == NULL || (dir != 0 && dir != 1)) {
        return;
    }
    int buf;
    for (int i = 1; i <=  n; i++)
    {
        buf = A[i];
        int j = i-1;
        while(j > 0 && (dir == 1 ? A[j] > buf : A[j] < buf)){
            A[j + 1] = A[j];
            j--;
        }
        A[j+1] = buf;
        PrintArray(A, n, dir);
    }

}
```

SelectionSort

```
//Vereb
```

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

void selectionSort(int a[], int n, int dir) {
    PrintArray(a, n, 1);

    for (int i = 1; i < n; i++) {
        int minidx = i;

        for (int j = i + 1; j <= n; j++) {
         if ((dir == 0 && a[j] > a[minidx]) || (dir == 1 && a[j] < a[minidx])) {
             minidx = j;
        }
        }

        if (minidx != i) {
            swap(&a[i], &a[minidx]);
        }

        PrintArray(a, n, i + 1);
    }
}

// Panik
void selectionSort(int a[], int n, int dir)
{
          for(int i = 1; i <= n; i++) {
                    int minidx = i;
                    for(int j = i + 1; j <= n; j++) {
                              if(a[j] < a[minidx] && dir) {
                                        minidx = j;
                              }
                              else if(a[j] > a[minidx] && !dir) {
                                        minidx = j;
                              }
                    }
                    PrintArray(a, n, dir);
                    int temp = a[i];
                    a[i] = a[minidx];
                    a[minidx] = temp;
          }
}

//Kurilak
void selectionSort(int a[], int n, int dir)
{
```

```
        int minidx, tmp;
    PrintArray(a,n,dir);

        if(dir == ASC){
      for(int i = 1; i < n; i++){
        minidx = i;
        for(int j = i + 1; j <= n; j++){
          if(a[j] < a[minidx]){
            minidx = j;
          }
        }
          tmp = a[minidx];
          a[minidx] = a[i];
          a[i] = tmp;
        PrintArray(a,n,dir);
      }
        }

        else{ //dir == DSC
      for (int i = 1; i <= n; i++) {
        minidx = i;
        for (int j = i + 1 ; j <= n; j++) {
          if (a[j] > a[minidx]) {
            minidx = j;
          }
        }
          tmp = a[minidx];
          a[minidx] = a[i];
          a[i] = tmp;
        PrintArray(a, n, dir);
      }
        }
}

//Soma
void selectionSort(int a[], int n, int dir) {
        PrintArray(a, n, dir);
        int i, j, minidx;
        for (i = 1; i < n; i++) {

                minidx = i;
                for (j = i + 1; j <= n; j++) {
                        if ((dir == ASC && a[j] < a[minidx]) || (dir == DSC && a[j] > a[minidx])) {
                                minidx = j;
                        }
                }
                Swap(a, i, minidx);
                PrintArray(a, n, j);
        }
}
```

```
void Swap(int a[], int i, int j){
        int temp = a[i];
        a[i] = a[j];
        a[j] = temp;
}
```

NASA HODINA

,

9.Cvičenie

BubbleSort

```
void bubbleSort(int a[], int n, int dir)
{
   if(a==NULL || n<=0 || (dir!=0 && dir!=1)){
      perror("Chyba");
   }
   int help_number=0;
   for(int i = 1; i <= n; i++) {
      for(int j = n; j >= i + 1; j--) {
         if ((dir == 1 && a[j - 1] > a[j]) || (dir == 0 && a[j] > a[j - 1])) {
            help_number=a[j-1];
            a[j-1]=a[j];
            a[j]=help_number;
         }
      }
      PrintArray(a,n,dir);
   }
}(Kucherenko Daniil)
```

InsertSort

-Opavský

```
void insertSort(int *A, int n, int dir)
{
    if(A == NULL || n < 1 || (dir != 0 && dir != 1)){
```

```
            printf("Chyba\n");
            return;
        }
        int value, j;
        for(int i=1; i<=n; i++){
            value = A[i];
            j = i-1;
            if(dir==1){
                while(j>=1 && A[j]>value){
                    A[j+1] = A[j];
                    j--;
                }
            }
            else{
                while(j>=1 && A[j]<value){
                    A[j+1] = A[j];
                    j--;
                }
            }
            A[j+1] = value;
            PrintArray(A, n, dir);
        }
    }
```

SelectionSort

Vida

```
void selectionSort(int a[], int n, int dir) {
    int temp;
    PrintArray(a, n);
    if (dir == 1) { //vzostupne
        int min;
        for (int i = 1; i < n; i++) {
            min = i;
            for (int j = i + 1; j <= n; j++) {
                if (a[j] < a[min]) {
                    min = j;
                }
            }
            temp = a[i];
            a[i] = a[min];
            a[min] = temp;
            PrintArray(a, n);
        }
    }
```

```
   if (dir == 0) { //zostupne
      int max;
      for (int i = 1; i < n; i++) {
         max = i;
         for (int j = i + 1; j <= n; j++) {
            if (a[j] > a[max]) {
               max = j;
            }
         }
         temp = a[i];
         a[i] = a[max];
         a[max] = temp;
         PrintArray(a, n);
      }
   }
}


// Mesarčík
void selectionSort(int a[], int n, int dir) {
   for (int i = 1; i <= n-1; i++) {
      int idx = i;
      for (int j = i+1; j <= n; j++) {
         if ( (dir == ASC && a[j] < a[idx]) || (dir == DSC && a[j] > a[idx]) ) {
            idx = j;
         }
      }
      int temp = a[i];
      a[i] = a[idx];
      a[idx] = temp;
      PrintArray(a, n, dir);
   }
}
```