Code(Selection Sort)

```
#include <iostream>
using namespace std;

void fillArray(int* a, int size, int& numberUsed);
void selectionSort(int a[], int n);
void swapValues(int& v1, int& v2);
int minSelect(int a[], int n);

const int NSIZE = 10;

int main()
{
  cout << "Sorting program" << endl;
  int sampleArray[NSIZE], numberUsed;
  fillArray(sampleArray, NSIZE, numberUsed);
  selectionSort(sampleArray, numberUsed);

  cout << "Sorted results" << endl;
  for (int index = 0; index < numberUsed; index++)
    cout << sampleArray[index] << " ";
  cout << endl;

  return 0;
}

void fillArray(int* a, int size, int& numberUsed)
{
  cout << " enter up to " << size << " nonnegative whole numbers" << endl
    << "Mark the end of the list with a negative number" << endl;
  int next, index = 0;
  cin >> next;
  while ((next >= 0) && (index < size))
  {
    a[index] = next;
    index++;
    cin >> next;
  }
  numberUsed = index;
}

void selectionSort(int a[], int n)
{

  int last(n - 1);
  int minPos;
  while (last > 0)
  {
    minPos = minSelect(a, last + 1);
    swapValues(a[minPos], a[last]);
    last--;
  }
}

void swapValues(int& v1, int& v2)
{
  int temp;
  temp = v1;
  v1 = v2;
  v2 = temp;

}

int minSelect(int a[], int n)
{
  int minPos(0), currentPos(1);
  while (currentPos < n) {
    // Invariant: a[maxPos] >= a[0] ... a[currentPos-1]
    if (a[currentPos] < a[minPos])
      minPos = currentPos;
    currentPos++;
  }
  return minPos;
}
```