**ALGORITHM**

**201133216**

**정유석**

Some people think that the bigger an elephant is,

the smarter it is. To disprove this, you want to

analyze a collection of elephants and place as

large a subset of elephants as possible into a

sequence whose weights are increasing but IQ’s

are decreasing.

**CODE**

#include<stdio.h>

#include<stdlib.h>

#define MAX\_WEIGHT 10000

#define MAX\_ELEPHANT 1000

typedef struct { //Structure is store the information of Elephant

int weight, IQ, number;

}Elephant;

typedef struct{ //It store the Number of Elephant

int num;

}Number;

Elephant ele[MAX\_ELEPHANT],ele\_find[MAX\_ELEPHANT][MAX\_ELEPHANT]; //ele is main information and ele\_find is order of Weight Increase, IQ decrease

Number num[MAX\_ELEPHANT]; //Number of Elephant

void main(){

int cnt,i,j,k=0, min\_w, min\_i, min\_n, num\_e; //cnt is number of elephant, i-j loop value, k is using number of elephant find, min\_w-n-e-i is using find minimum value

int temp\_w, temp\_i, temp\_n, temp\_IQ; //temporary value

int length[MAX\_ELEPHANT]; //Each length

int p, find\_l, find\_n;

printf("Enter the number of elepbant : ");

scanf("%d",&cnt); //Receive number of elephant

if(cnt < 2 || cnt > MAX\_ELEPHANT){ //Exception

printf("WRONG INPUT! -Number-\n\n");

exit(0);

}

for(i=0; i<cnt; i++){

printf("Input weight of %dst elephant : ",i+1);

scanf("%d",&ele[i].weight);

if(ele[i].weight < 2 || ele[i].weight > MAX\_WEIGHT){ //Exception

printf("WRONG INPUT! -Weight-\n\n");

exit(0);

}

printf("Input IQ of %dst elephant : ",i+1);

scanf("%d",&ele[i].IQ);

if(ele[i].IQ < 1 || ele[i].IQ > 10000){ //Exception

printf("WRONG INPUT! -IQ-\n\n");

exit(0);

}

ele[i].number = i+1;

}

//-- array is ordered small weight

for(i=0; i<cnt; i++){ //Use the selection sort

min\_w = ele[i].weight;

min\_i = ele[i].IQ;

min\_n = ele[i].number;

k=i;

for(j=i; j<cnt; j++){

if(min\_w > ele[j].weight){ //find the minmum of weight and insert to array order

min\_w = ele[j].weight;

min\_i = ele[j].IQ;

min\_n = ele[j].number;

k = j;

}

else

continue;

}

if(k==i)

continue;

else{

temp\_w = ele[i].weight;

temp\_i = ele[i].IQ;

temp\_n = ele[i].number;

ele[i].weight = min\_w;

ele[i].IQ = min\_i;

ele[i].number = min\_n;

ele[k].IQ = temp\_i;

ele[k].number = temp\_n;

ele[k].weight = temp\_w;

}

}

for(i=0; i<cnt; i++){

printf("%d \n",ele[i].weight);

}

for(i=0; i<cnt; i++){ //and find the order -- increase weight .. decrease weight

temp\_IQ = ele[i].IQ;

length[i] = 1;

p=0;

ele\_find[i][p].number = ele[i].number;

for(j=i; j<cnt; j++){

if(temp\_IQ > ele[j].IQ){

length[i] ++;

temp\_IQ = ele[j].IQ;

p++;

ele\_find[i][p].number = ele[j].number;

}

}

}

find\_l = length[0];

for(i=0; i<cnt; i++){

if( find\_l < length[i]){ //Find the longer length

find\_l = length[i];

find\_n = i;

}

}

printf("Lengh : %d \n",find\_l);

i=0;

while(ele\_find[find\_n][i].number){ //Display

printf(" Number : %d \n",ele\_find[find\_n][i].number);

i++;

}

}

**PRINT**

