

```
//An implementation of Algorithms 5.3, 5.4 and 5.5 (Heap Sort)
//written by Kazutoshi Ando (Shizuoka University)

#include <stdio.h>
#define n 10
#define SWAP(x,y) {typeof(x) tmp; tmp=x; x=y; y=tmp;} // xとyの値を交換する

int size=0; //ヒープのサイズを表す変数

void show_array(int A[], int l) { //配列の中身を表示する関数
    int i;
    printf("[");
    for (i=0;i<l;i++) {
        printf(" %2d",A[i]);
        if(i<l-1) printf(",");
    }
    printf("]\n");
}

void push_heap(int T[],int x){ // アルゴリズム 5.3
    int k;

    size++;
    T[size] = x;
    k= size;
    while(T[k]>T[k/2] && k>1) { // 親子の値を比較
        SWAP(T[k],T[k/2]); // 親の値が小さい場合は値を交換
        k=k/2;
    }
}

int delete_maximum(int T[]) { // アルゴリズム 5.4
    int k,big,ret_val;

    ret_val= T[1];
    T[1] = T[size];
    T[size]=-1;
    size--;
    k=1;
    while (2*k <= size) { // 子をもつかどうか判定
        if (2*k==size) { // 子が1つの場合
            if(T[k] < T[2*k]){ // 親子の値を比較
                SWAP(T[k],T[2*k]); // 親の値が小さい場合は値を交換
                k = 2*k;
            } else {
                break; // while文の外へ出る
            }
        } else { // 子供が2つの場合
            if(T[2*k] > T[2*k+1]) { //大きいデータをもつ子をもつ
                big = 2*k;
            } else {
                big = 2*k+1;
            }
            if(T[k]<T[big]) { // 親子の値を比較
                SWAP(T[k],T[big]); //親の値が小さい場合は値を交換
                k=big;
            } else {
                break; // while文の外へ出る
            }
        }
    }
    return ret_val;
}
```

```
main(){
  int D[n]={17,39,1,9,5,24,2,11,23,6};
  int T[n+1]; //ヒープ
  int i;

  for(i=0;i<n+1;i++) T[i]=-1;
  // T[i] == -1 のとき , T[i]にはデータ格納されていなと考える

  show_array(D,10);
  for(i=0;i<n;i++) {
    push_heap(T,D[i]);
    //show_array(T,11);
  }
  for(i=n-1;i>=0;i--) {
    D[i]=delete_maximum(T);
    //show_array(T,11);
  }
  show_array(D,10);
}
```