Jenish Patel

Prof. Ken

CS350

09/15/2020

Title: HW2

#include <stdio.h>

#include <string.h>

struct student {

int id;

char name[16];

float gpa;

};

typedef struct student STUDENT;

void getStudent(STUDENT\* pstu) {

printf("Enter id, name, gpa:\n");

scanf("%d", &(pstu->id));

scanf("%s", &(pstu->name));

scanf("%f", &(pstu->gpa));

}

void printStudent(STUDENT stu) {

printf("You have entered:\n");

printf("id: %d\n", stu.id);

printf("name: %s\n", stu.name);

printf("gpa: %.1f\n", stu.gpa);

}

void copyStudent(STUDENT\* original, STUDENT\* another ) {

    another->id = original ->id;

    strcpy(original->name, another ->name);

    another->gpa = original->gpa;

}

int main() {

STUDENT stu;

getStudent(&stu);

printStudent(stu);

return 0;

STUDENT newStu;

copyStudent(&stu, &newStu);

printStudent(newStu);

return 0;

}

Output:

Enter id, name, gpa:

45

Jenish

3

You have entered:

id: 45

name: Jenish

gpa: 3.0

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#define N 10

// fill in the list with random numbers from 0 to 99

void getData(int\* list, int size) {

for (int i = 0; i < size; i++) {

list[i] = rand() % 100;

}

}

// print the entire list

void printData(int\* list, int size) {

for (int i = 0; i < size; i++) {

printf("%d\t", list[i]);

}

printf("\n");

}

// find the maximum value in the list

int findMax(int\* list, int size) {

int max = list[0];

for (int i = 1; i < size; i++) {

if (list[i] > max) {

max = list[i];

}

}

return max;

}

// find the index of the target in the list

int search(int\* list, int size, int target) {

for (int i = 0; i < size; i++) {

if (list[i] == target) {

return i; // found

}

}

return -1; // not found

}

// reverse the list

void reverse(int\* list, int size) {

int temp;

for (int i = 0; i < size / 2; i++) {

temp = list[i];

list[i] = list[size - i - 1];

list[size - i - 1] = temp;

}

}

// sort the list

void sort(int\* list, int size) {

int temp;

for (int pass = 0; pass < size - 1; pass++) {

for (int i = 0; i < size - pass - 1; i++) {

if (list[i] > list[i+1]) {

temp = list[i];

list[i] = list[i+1];

list[i+1] = temp;

}

}

}

}

void removeNegative(int\* list, int\* size) {

    int i;

    int count=0;

    if(list==-1)

        {

            size[i]=size[i-1];

            count--;

            i=20;

        }

        else if(list<-1)

        {

            size[i]=size[i-1];

            count--;

        }

    }

int main() {

int list[N];

srand(time(0));

// fill the list

getData(list, N);

printData(list, N);

// find the max in the list

int max = findMax(list, N);

printf("the maximum value is %d\n", max);

printf("the index of the max value is %d\n", search(list, N, max));

// reverse the list

reverse(list, N);

printData(list, N);

// sort the list

sort(list, N);

printData(list, N);

removeNegative(list, sizeof(int));

printData(list, 5);

return 0;

}

Output:

54 0 33 20 22 76 84 73 63 84

the maximum value is 84

the index of the max value is 6

84 63 73 84 76 22 20 33 0 54

0 20 22 33 54 63 73 76 84 84

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#define N 10

// fill in the list with random numbers from 0 to 99

void getData(int\* list, int size) {

for (int i = 0; i < size; i++) {

list[i] = rand() % 100;

}

}

// print the entire list

void printData(int\* list, int size) {

for (int i = 0; i < size; i++) {

printf("%d\t", list[i]);

}

printf("\n");

}

// a simple function for comparing two integers

int compare (const void \* a, const void \* b) {

return ( \*(int\*)a - \*(int\*)b );

}

// rotate the list by x elements

void rotate(int\* list, int size, int x) {

for (int n = 0; n < x; n++) {

int temp = list[size - 1];

for (int i = size - 1; i > 0; i--) {

list[i] = list[i-1];

}

list[0] = temp;

}

}

// merge two sorted lists

int\* merge(int\* list1, int size1, int\* list2, int size2) {

int i, j, k;

int\* list3 = (int\*) malloc(sizeof(int) \* (size1 + size2));

i = j = k = 0;

while (i < size1 && j < size2) {

if (list1[i] < list2[j]) {

list3[k++] = list1[i++];

} else {

list3[k++] = list2[j++];

}

}

while (i < size1) {

list3[k++] = list1[i++];

}

while (j < size2) {

list3[k++] = list2[j++];

}

return list3;

}

void shuffle(int\* list, int size){

    int temp = \*list;

    \*list = size;

    size = temp;

}

int main() {

int list[N];

srand(time(0));

// fill the list

getData(list, N);

printData(list, N);

// rotate the list 2 times

rotate(list, N, 2);

printData(list, N);

// create two lists

int list1[5];

int list2[6];

// fill the lists

getData(list1, 5);

qsort(list1, 6, sizeof(int), compare);

printData(list1, 5);

getData(list2, 6);

qsort(list2, 6, sizeof(int), compare);

printData(list2, 6);

// merge list1 and list2 into list3

int\* list3 = merge(list1, 5, list2, 6);

printData(list3, 11);

shuffle(list, sizeof(int));

printData(list1, 5);

return 0;

}

Output:

15 14 58 1 46 60 35 58 99 12

99 12 15 14 58 1 46 60 35 58

0 23 50 65 77

23 38 71 78 93 98

0 23 23 38 50 65 71 77 78 93 98

0 23 50 65 77

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

struct Book // struct Book

{

    char author[32];

    char title[32];

    char publisher[32];

    int year;

    char ISBN[15];

};

// readBookRecord() function

int main()

{

    struct Book book;

    strcpy(book.author, "Jenish");

    strcpy(book.title, "Empirial square");

    strcpy(book.publisher, "Newyorkbooks");

    book.year = 2014;

    strcpy(book.ISBN, "98563247");

    printf("Book Record\n");

    printf("Book Author: %s\n", book.author);

    printf("Book Title: %s\n", book.title);

    printf("Book Publisher: %s\n", book.publisher);

    printf("Book Publishing Year: %d\n", book.year);

    printf("Book ISBN: %s\n", book.ISBN);

}

Output:

Book Record

Book Author: Jenish

Book Title: Empirial square

Book Publisher: Newyorkbooks

Book Publishing Year: 2014

Book ISBN: 98563247

#include<stdio.h>

struct locator

{

char Landlord[50];

char Address[100];

int Bedrooms;

float Price;

};

typedef struct locator Locator;

void insert(Locator\* loc){

    printf("Please Entr Landlord Name: ");

    scanf("%s", (loc->Landlord));

    printf("Please Entr Address: ");

    scanf("%s", (loc->Address));

    printf("Please Enter count of the Bedroom: ");

    scanf("%d", &(loc->Bedrooms));

    printf("Please Enter The Price: ");

    scanf("%lf", &(loc->Price));

}

void print(Locator l) {

printf("You have entered:\n");

printf("Name of the Landlord is: %s\n", l.Landlord);

printf("Address of the plot: %s\n", l.Address);

printf("Here is the count of the bedroom: ", l.Bedrooms);

printf("Price of Plot : %lf\n", l.Price);

}

int main(){

    Locator l;

    insert(&l);

    print(l);

    return 0;

}

Output:

Please Entr Landlord Name: Jenish

Please Entr Address: Laklen\_dr

Please Enter count of the Bedroom: 4

Please Enter The Price: 250000

You have entered:

Name of the Landlord is: Jenish

Address of the plot: Laklen\_dr

Here is the count of the bedroom: 4

Price of Plot : 250000.000000