**Workshop 7**

**Tên: Trần Nguyễn Quốc Cường**

Tiếp thu 90%, điểm slot: 9đ

**Problem 1: (3 marks) Managing a list of student names**

**#include <stdio.h>**

**#include <string.h>**

**const int STRLEN = 100;**

**const int NUMBEROFSTUDENTNAME = 100;**

**void add(char studentNameList[][STRLEN], char studentName[], short\* studentNameListSize) {**

**(\*studentNameListSize) += 1;**

**short currNameIndex = (\*studentNameListSize) - 1;**

**char\* token = NULL;**

**token = strtok(studentName, " ");**

**strcpy(studentNameList[currNameIndex], strupr(token));**

**while (true) {**

**token = strtok(NULL, " ");**

**if (!token) {**

**return;**

**}**

**strcat(studentNameList[currNameIndex], " ");**

**strcat(studentNameList[currNameIndex], strupr(token));**

**}**

**}**

**void deleteStudent(char studentNameList[][STRLEN], short\* studentNameListSize, int inputIndex) {**

**if (inputIndex == (\*studentNameListSize) - 1) {**

**strcpy(studentNameList[inputIndex], "");**

**} else {**

**short currNameIndex;**

**for (currNameIndex = inputIndex; currNameIndex < (\*studentNameListSize); ++currNameIndex) {**

**strcpy(studentNameList[currNameIndex], studentNameList[currNameIndex + 1]);**

**}**

**}**

**(\*studentNameListSize) -= 1;**

**}**

**int search(char studentNameList[][STRLEN], short studentNameListSize, char studentName[]) {**

**short currNameIndex;**

**int check = -1;**

**for (currNameIndex = 0; currNameIndex < studentNameListSize; ++currNameIndex) {**

**if (strcmp(studentNameList[currNameIndex], studentName) == 1) {**

**check = 1;**

**}**

**}**

**return check;**

**}**

**void sortStudent(char studentNameList[][STRLEN], short studentNameListSize) {**

**char swapStr[STRLEN] = "\0";**

**int idx1;**

**int idx2;**

**for (idx1 = 0; idx1 < studentNameListSize - 1; ++idx1) {**

**for (idx2 = idx1 + 1; idx2 < studentNameListSize; ++idx2) {**

**if (strcmp(studentNameList[idx1], studentNameList[idx2]) > 0) {**

**strcpy(swapStr, studentNameList[idx1]);**

**strcpy(studentNameList[idx1], studentNameList[idx2]);**

**strcpy(studentNameList[idx2], swapStr);**

**}**

**}**

**}**

**}**

**void printList(char studentNameList[][STRLEN], short studentNameListSize) {**

**short currNameIndex;**

**printf("\n[");**

**for (currNameIndex = 0; currNameIndex < studentNameListSize - 1; ++currNameIndex) {**

**printf("%s, ", studentNameList[currNameIndex]);**

**}**

**printf("%s]\n", studentNameList[currNameIndex]);**

**}**

**int main() {**

**char studentNameList[NUMBEROFSTUDENTNAME][STRLEN];**

**short studentNameListSize = 0;**

**int choice;**

**printf("--- MENU ---\n");**

**printf("1. Add a student\n");**

**printf("2. Remove a student\n");**

**printf("3. Search a student\n");**

**printf("4. Print the list in ascending order\n");**

**printf("5. Quit\n");**

**do {**

**printf("Your choice : ");**

**scanf("%d", &choice);**

**switch (choice) {**

**case 1: {**

**char studentName[STRLEN] = "\0";**

**printf("Student name: ");**

**scanf("%s", studentName);**

**add(studentNameList, studentName, &studentNameListSize);**

**printf("Add successfully !!\n");**

**break;**

**}**

**case 2: {**

**if (studentNameListSize == 0) {**

**printf("There is no students yet\n");**

**break;**

**}**

**long inputIndex;**

**printf("Input index: ");**

**scanf("%d", &inputIndex);**

**if (inputIndex < 0 || inputIndex >= studentNameListSize) {**

**printf("There is no such student\n");**

**} else {**

**deleteStudent(studentNameList, &studentNameListSize, inputIndex);**

**}**

**break;**

**}**

**case 3: {**

**if (studentNameListSize == 0) {**

**printf("There is no students yet\n");**

**break;**

**}**

**char studentName[STRLEN] = "\0";**

**printf("Student name: ");**

**scanf("%s", studentName);**

**if (search(studentNameList, studentNameListSize, studentName)) {**

**printf("Student is found\n");**

**} else {**

**printf("Student is not found\n");**

**}**

**break;**

**}**

**case 4: {**

**if (studentNameListSize == 0) {**

**printf("There is no students yet\n");**

**break;**

**}**

**sortStudent(studentNameList, studentNameListSize);**

**printList(studentNameList, studentNameListSize);**

**break;**

**}**

**}**

**} while ((choice > 0) && (choice < 5));**

**return 0;**

**}**

**Problem 2: (3 marks) Managing a parallel arrays**

**#include <stdio.h>**

**#include <string.h>**

**const int MAX\_N = 100;**

**int main() {**

**char code[100][8];**

**char name[100][20];**

**double salary[100];**

**double allowance[100];**

**int choice;**

**int employeeCount = 0;**

**printf("--- MENU ---\n");**

**printf("1. Adding a new employee\n");**

**printf("2. Find data about employees using a name inputted\n");**

**printf("3. Remove an employee based on a code inputted\n");**

**printf("4. Print the list in descending order based on salary + allowance\n");**

**printf("5. Quit\n");**

**do {**

**printf("Your choice: ");**

**scanf("%d", &choice);**

**switch (choice) {**

**case 1: {**

**short i;**

**i = (++employeeCount) - 1;**

**printf("New employee code: ");**

**scanf("%s", code[i]);**

**code[i][strcspn(code[i], "\n")] = '\0';**

**printf("New employee name: ");**

**scanf("%s", name[i]);**

**name[i][strcspn(name[i], "\n")] = '\0';**

**printf("New employee salary: ");**

**scanf("%lf", &salary[i]);**

**printf("New employee allowance: ");**

**scanf("%lf", &allowance[i]);**

**break;**

**}**

**case 2: {**

**if (employeeCount == 0) {**

**printf("There is no employee\n");**

**} else {**

**char inputtedName[20] = "\0";**

**int i;**

**int check = -1;**

**printf("Employee name: ");**

**scanf("%s", inputtedName);**

**inputtedName[strcspn(inputtedName, "\n")] = '\0';**

**for (i = 0; i < employeeCount; ++i) {**

**if (!strcmp(inputtedName, name[i]) == 1) {**

**check = 1;**

**break;**

**}**

**}**

**if (check == 1) {**

**printf("Employee data:\n");**

**printf("Code: %s\n", code[i]);**

**printf("Name: %s\n", name[i]);**

**printf("Salary: %lf\n", salary[i]);**

**printf("Allowance: %lf\n", allowance[i]);**

**} else {**

**printf("Inputted employee is not found\n");**

**}**

**}**

**break;**

**}**

**case 3: {**

**if (employeeCount == 0) {**

**printf("There is no employee\n");**

**} else {**

**char inputtedCode[20] = "\0";**

**int check = -1;**

**short i;**

**short inputtedNameIdx;**

**printf("\nEmployee code: ");**

**scanf("%s", inputtedCode);**

**inputtedCode[strcspn(inputtedCode, "\n")] = '\0';**

**for (i = 0; i <= employeeCount; ++i) {**

**if (strcmp(inputtedCode, code[i]) == 1) {**

**check = 1;**

**break;**

**}**

**}**

**if (check == 1) {**

**for (i = 0; i <= employeeCount; ++i) {**

**if (strcmp(inputtedCode, code[i]) == 1) {**

**inputtedNameIdx = i;**

**break;**

**}**

**}**

**if (inputtedNameIdx == employeeCount - 1) {**

**strcpy(code[inputtedNameIdx], "");**

**strcpy(name[inputtedNameIdx], "");**

**salary[inputtedNameIdx] = 0;**

**allowance[inputtedNameIdx] = 0;**

**} else {**

**for (i = inputtedNameIdx; i < employeeCount - 1; ++i) {**

**strcpy(code[i], code[i + 1]);**

**}**

**strcpy(code[i], "");**

**for (i = inputtedNameIdx; i < employeeCount - 1; ++i) {**

**strcpy(name[i], name[i + 1]);**

**}**

**strcpy(name[i], "");**

**for (i = inputtedNameIdx; i < employeeCount - 1; ++i) {**

**salary[i] = salary[i + 1];**

**}**

**salary[i] = 0;**

**for (i = inputtedNameIdx; i < employeeCount - 1; ++i) {**

**allowance[i] = allowance[i + 1];**

**}**

**allowance[i] = 0;**

**}**

**--employeeCount;**

**printf("Removed sucessfully\n");**

**} else {**

**printf("No employee is found\n");**

**}**

**}**

**break;**

**}**

**case 4: {**

**if (employeeCount == 0) {**

**printf("There is no employee\n");**

**} else {**

**short i;**

**short j;**

**double dSwap;**

**char sSwap[20] = "\0";**

**for (i = 0; i < employeeCount - 1; ++i) {**

**for (j = i + 1; j < employeeCount; ++j) {**

**if (salary[i] < salary[j] || allowance[i] < allowance[j]) {**

**strcpy(sSwap, code[i]);**

**strcpy(code[i], code[j]);**

**strcpy(code[j], sSwap);**

**strcpy(sSwap, name[i]);**

**strcpy(name[i], name[j]);**

**strcpy(name[j], sSwap);**

**dSwap = salary[i];**

**salary[i] = salary[j];**

**salary[j] = dSwap;**

**dSwap = allowance[i];**

**allowance[i] = allowance[j];**

**allowance[j] = dSwap;**

**}**

**}**

**}**

**for (i = 0; i < employeeCount; ++i) {**

**printf("Employee No. %d: \n", i + 1);**

**printf("Code: %s\n", code[i]);**

**printf("Name: %s\n", name[i]);**

**printf("Salary: %lf\n", salary[i]);**

**printf("Allowance: %lf\n", allowance[i]);**

**}**

**}**

**break;**

**}**

**}**

**} while ((choice > 0) && (choice < 5));**

**return 0;**

**}**

**Problem 3: (4 marks) Managing a parallel arrays**

**#include <stdio.h>**

**#include <string.h>**

**const int MAX\_N = 100;**

**int main() {**

**char name[100][20];**

**char make[100][20];**

**int volume[100];**

**int price[100];**

**int duration[100];**

**int choice;**

**short listCount = 0;**

**printf("--- MENU ---\n");**

**printf("1. Adding a new soft drink\n");**

**printf("2. Printing out items which belong to a known make\n");**

**printf("3. Printing out items whose volumes are between v1 and v2 (integers)\n");**

**printf("4. Printing the list in ascending order based on volumes then prices\n");**

**printf("5. Quit\n");**

**do {**

**printf("Your choice: ");**

**scanf("%d", &choice);**

**switch (choice) {**

**case 1: {**

**int i;**

**i = (++listCount) - 1;**

**printf("\nNew drink name: ");**

**scanf("%s", name);**

**name[i][strcspn(name[i], "\n")] = '\0';**

**printf("New drink make: ");**

**scanf("%s", make);**

**make[i][strcspn(make[i], "\n")] = '\0';**

**printf("New drink volume: ");**

**scanf("%d", &volume[i]);**

**printf("New drink price: ");**

**scanf("%d", &price[i]);**

**printf("New drink duration: ");**

**scanf("%d", &duration[i]);**

**break;**

**}**

**case 2: {**

**if (listCount == 0) {**

**printf("There is no drink\n");**

**} else {**

**char inputtedMake[20] = "\0";**

**short i;**

**printf("\nInput drink make: ");**

**scanf("%s", inputtedMake);**

**inputtedMake[strcspn(inputtedMake, "\n")] = '\0';**

**printf("Items belonging to a known make: \n");**

**for (i = 0; i < listCount; ++i) {**

**if (!strcmp(inputtedMake, make[i])) {**

**printf("Drink %d:\n", i + 1);**

**printf("- Name: %s\n", name[i]);**

**printf("- Make: %s\n", make[i]);**

**printf("- Volume: %d\n", volume[i]);**

**printf("- Price: %d\n", price[i]);**

**printf("- Duration: %d\n", duration[i]);**

**}**

**}**

**}**

**break;**

**}**

**case 3: {**

**if (listCount == 0) {**

**printf("No drinks in the list\n");**

**} else {**

**int v1;**

**int v2;**

**int i;**

**printf("Input v1: ");**

**scanf("%d", &v1);**

**printf("Input v2: ");**

**scanf("%d", &v2);**

**for (i = 0; i < listCount; ++i) {**

**if (volume[i] >= v1 && volume[i] <= v2) {**

**printf("Drink No. %d:\n", i + 1);**

**printf("- Name: %s\n", name[i]);**

**printf("- Make: %s\n", make[i]);**

**printf("- Volume: %d\n", volume[i]);**

**printf("- Price: %d\n", price[i]);**

**printf("- Duration: %d\n", duration[i]);**

**}**

**}**

**}**

**break;**

**}**

**case 4: {**

**if (listCount == 0) {**

**printf("No drinks in the list\n");**

**} else {**

**short i;**

**short j;**

**int iSwap;**

**char sSwap[20] = "\0";**

**for (i = 0; i < listCount - 1; ++i) {**

**for (j = i + 1; j < listCount; ++j) {**

**if (volume[i] > volume[j] || price[i] > price[j]) {**

**strcpy(sSwap, name[i]);**

**strcpy(name[i], name[j]);**

**strcpy(name[j], sSwap);**

**strcpy(sSwap, make[i]);**

**strcpy(make[i], make[j]);**

**strcpy(make[j], sSwap);**

**iSwap = volume[i];**

**volume[i] = volume[j];**

**volume[j] = iSwap;**

**iSwap = price[i];**

**price[i] = price[j];**

**price[j] = iSwap;**

**iSwap = duration[i];**

**duration[i] = duration[j];**

**duration[j] = iSwap;**

**}**

**}**

**}**

**for (i = 0; i < listCount; ++i) {**

**printf("Drink No. %d:\n", i + 1);**

**printf("- Name: %s\n", name[i]);**

**printf("- Make: %s\n", make[i]);**

**printf("- Volume: %d\n", volume[i]);**

**printf("- Price: %d\n", price[i]);**

**printf("- Duration: %d\n", duration[i]);**

**}**

**}**

**break;**

**}**

**}**

**} while ((choice > 0) && (choice < 5));**

**return 0;**

**}**