***Завдання 12(№13)***

*#include <stdio.h>*

*#include <stdlib.h>*

*#include <time.h>*

*#include <math.h>*

*#include <string.h>*

*/\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/*

*typedef struct Node {*

*char value;*

*struct Node \*next;*

*};*

*int main(int argc, char \*argv[]) {*

*int listSize = 0, uniqueAmount = 0, i = 0;*

*Node \*head = NULL, \*iterator = NULL, \*tmpPtr = NULL, \*uniqueHead = NULL;*

*printf("Input N:\n");*

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

*while (listSize < 1) {*

*printf("Size of list should be greater than 0. Try again.\n");*

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

*}*

*printf("Type list of %d characters:\n", listSize);*

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

*tmpPtr = (Node\*) malloc(sizeof(Node));*

*tmpPtr->next = NULL;*

*scanf("%s", &(tmpPtr->value));*

*if (head == NULL) {*

*head = tmpPtr;*

*} else {*

*iterator->next = tmpPtr;*

*}*

*iterator = tmpPtr;*

*}*

*iterator = head;*

*while (iterator != NULL) {*

*if (uniqueHead == NULL) {*

*uniqueHead = (Node\*) malloc(sizeof(Node));*

*uniqueHead->next = NULL;*

*uniqueHead->value = iterator->value;*

*uniqueAmount++;*

*iterator = iterator->next;*

*continue;*

*}*

*tmpPtr = uniqueHead;*

*while (tmpPtr != NULL) {*

*if (tmpPtr->value == iterator->value || tmpPtr->next == NULL) { break; }*

*tmpPtr = tmpPtr->next;*

*}*

*if (tmpPtr->value != iterator->value) {*

*tmpPtr->next = (Node\*) malloc(sizeof(Node));*

*tmpPtr->next->value = iterator->value;*

*tmpPtr->next->next = NULL;*

*uniqueAmount++;*

*}*

*iterator = iterator->next;*

*}*

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

*while (head != NULL) {*

*printf("%c ", head->value);*

*iterator = head;*

*head = head->next;*

*free(iterator);*

*}*

*printf("\n\n------Unique Characters List-------\n");*

*while (uniqueHead != NULL) {*

*printf("%c ", uniqueHead->value);*

*iterator = uniqueHead;*

*uniqueHead = uniqueHead->next;*

*free(iterator);*

*}*

*printf("\n\nAmount of unique character: %d \n", uniqueAmount);*

*return 0;*

*}*