פרויקט מספר:

2

שאלה מספר 1

פולינום

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include <string.h>

//סטראקטים

typedef struct node {

int mekadem;

int hezka;

struct node\* next;

} node;

//פונקצה אשר מחליפה את הערכים של שני פולינומים

void swap(struct node\* a, struct node\* b)

{

node temp;

temp.mekadem = a->mekadem;

temp.hezka = a->hezka;

a->mekadem = b->mekadem;

a->hezka = b->hezka;

b->mekadem = temp.mekadem;

b->hezka = temp.hezka;

}

//פונקיצה אשר ממינת את הפולינומים לפי החזקה

void bubble\_nodes\_sort(struct node\* start)

{

int swapped, i;

node\* ptr1;

node\* lptr = NULL;

/\* Checking for empty list \*/

if (start == NULL)

return;

do

{

swapped = 0;

ptr1 = start;

while (ptr1->next != lptr)

{

if (ptr1->hezka < ptr1->next->hezka)

{

swap(ptr1, ptr1->next);

swapped = 1;

}

ptr1 = ptr1->next;

}

lptr = ptr1;

} while (swapped);

}

//פונקציה שמדפיסה פולינום

void print\_nodes(node\* list, const char name\_of\_list[])

{

printf("linked list %s is: ", name\_of\_list);

if (list == NULL)

printf("NULL ");

else

{

while (list)

{

printf("%d\*x^%d ", list->mekadem, list->hezka);

list = list->next;

}

printf("\n");

}

}

//פןנקציה שהופכת סטרינג לפולינום

void str\_to\_nodes(node\* list, char strX[50])

{

char str[50];

strcpy(str, strX);

int flag = 0;

node\* p = list;

node\* p\_next = NULL;

char str\_mekadem[50];

char str\_hezka[50];

int i = 0;

while (str[0])

{

while (str[i])

{

if (str[i] != ' ')

{

//העתקה

str\_mekadem[i] = str[i];

}

else

{

break;

}

i++;

str\_mekadem[i] = 0;

}

strcpy(str, (str + (i + 1)));

i = 0;

while (1)

{

if (str[i] == ' ' || str[i] == 0)

{

str\_hezka[i] = 0;

if (str[i] == ' ')

strcpy(str, (str + (i + 1)));

else

str[0] = 0;

break;

}

else

{

//העתקה

str\_hezka[i] = str[i];

}

i++;

str\_hezka[i] = 0;

}

if (list != p)

{

p\_next = (node\*)malloc(sizeof(node));

p\_next->mekadem = atoi(str\_mekadem);

p\_next->hezka = atoi(str\_hezka);

p\_next->next = NULL;

p->next = p\_next;

p = p->next;

p\_next = NULL;

}

else if (flag == 0)

{

p->mekadem = atoi(str\_mekadem);

p->hezka = atoi(str\_hezka);

p->next = NULL;

flag = 1;

}

else

{

p\_next = (node\*)malloc(sizeof(node));

p\_next->mekadem = atoi(str\_mekadem);

p\_next->hezka = atoi(str\_hezka);

p\_next->next = NULL;

p->next = p\_next;

p = p->next;

p\_next = NULL;

}

i = 0;

}

}

//פונקציה אשר גוזרת פולינום

void cut(node\* list)

{

if (list != NULL)

{

node\* p = list;

while (p != NULL)

{

p->mekadem \*= p->hezka;

p->hezka--;

if (p->next != NULL)

if (p->next->hezka == 0)

{

p->next = NULL;

break;

}

p = p->next;

}

}

}

//פונקציה אשר ממזגת פןלינומים בעליחזקה זהה

void merge\_same\_nodes(node\* list)

{

node\* p1 = list, \* p2, \* p2\_next;

while (p1)

{

p2 = p1;

p2\_next = p2->next;

while (p2\_next)

{

if (p1->hezka == p2\_next->hezka)

{

p1->mekadem += p2\_next->mekadem;

p2->next = p2\_next->next;

free(p2\_next);

p2\_next;

p2\_next = p2->next;

}

else

{

p2 = p2->next;

p2\_next = p2->next;

}

}

p1 = p1->next;

}

}

//פעולנ המחליפה בסטרינג קו תחתון ברווח

void kav\_tahton\_to\_revah(char str[])

{

int len = strlen(str);

for (int i = 0; i < len; i++)

if (str[i] == '\_')

str[i] = ' ';

}

//פונקציה אשר מכפילה פולינומים

void kaful(node\* list1, node\* list2, node\* final\_list)

{

node\* p = final\_list, \* p\_next, \* p\_list2;

int flag = 0;

while (list1)

{

p\_list2 = list2;

while (p\_list2)

{

if (final\_list != p)

{

p\_next = (node\*)malloc(sizeof(node));

p\_next->mekadem = list1->mekadem \* p\_list2->mekadem;

p\_next->hezka = list1->hezka + p\_list2->hezka;

p\_next->next = NULL;

p->next = p\_next;

p = p->next;

p\_next = NULL;

}

else if (flag == 0)

{

p->mekadem = list1->mekadem \* p\_list2->mekadem;

p->hezka = list1->hezka + p\_list2->hezka;

p->next = NULL;

flag = 1;

}

else

{

p\_next = (node\*)malloc(sizeof(node));

p\_next->mekadem = list1->mekadem \* p\_list2->mekadem;

p\_next->hezka = list1->hezka + p\_list2->hezka;

p\_next->next = NULL;

p->next = p\_next;

p = p->next;

p\_next = NULL;

}

p\_list2 = p\_list2->next;

}

list1 = list1->next;

}

merge\_same\_nodes(final\_list);

bubble\_nodes\_sort(final\_list);

}

void main()

{

char str1[50], str2[50], str\_plus[100];

node\* head1 = (node\*)malloc(sizeof(node)), \* head2 = (node\*)malloc(sizeof(node))

, \* head1\_plus\_head2 = (node\*)malloc(sizeof(node)),

\* head1\_kaful\_head2 = (node\*)malloc(sizeof(node));

printf("please enter polynom\n");

fgets(str1, 100, stdin);

kav\_tahton\_to\_revah(str1);

printf("please enter polynom\n");

fgets(str2, 100, stdin);

kav\_tahton\_to\_revah(str1);

str\_to\_nodes(head1, str1);

str\_to\_nodes(head2, str2);

print\_nodes(head1, "head1");

print\_nodes(head2, "head2");

//חיבור

strcpy(str\_plus, str1);

strcat(str\_plus, " ");

strcat(str\_plus, str2);

str\_to\_nodes(head1\_plus\_head2, str\_plus);

merge\_same\_nodes(head1\_plus\_head2);

bubble\_nodes\_sort(head1\_plus\_head2);

print\_nodes(head1\_plus\_head2, "head1\_plus\_head2");

//כפל

kaful(head1, head2, head1\_kaful\_head2);

print\_nodes(head1\_kaful\_head2, "head1\_kaful\_head2");

//גזירה 1

printf("cut:\n");

cut(head1\_kaful\_head2);

print\_nodes(head1\_kaful\_head2, "head1\_kaful\_head2");

//גזירה 2

printf("cut:\n");

cut(head1\_kaful\_head2);

print\_nodes(head1\_kaful\_head2, "head1\_kaful\_head2");

}

שאלה מספר 2

תפזורת

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <time.h>

#include <stdlib.h>

#include <string.h>

#define X 10

//פונקציה אשר מכניסה אותיות באופן רנדומלי ללוח

void fill\_board\_randomally(char arr[][X])

{

for (int i = 0; i < X; i++)

for (int a = 0; a < X; a++)

{

int x = (rand() % (122 + 1 - 97) + 97);

char c = x;

arr[i][a] = c;

}

}

//פונקציה אשר ממצאת את הלוח באופן ידני

void fill\_board\_by\_yourself(char arr[][X])

{

for (int i = 0; i < X; i++)

{

printf("\nline number %d:\n", i + 1);

for (int a = 0; a < X; a++)

{

char c;

scanf(" %c", &c);

arr[i][a] = c;

}

}

}

//פונקציה שמדפיסה את הלוח

void print\_board(char arr[][X])

{

for (int i = 0; i < X; i++)

{

for (int a = 0; a < X; a++)

printf(" %c", arr[i][a]);

printf("\n");

}

}

//פןנקציה שבודקת האם יש שוויון בין המילה הנתונה לבין המילה בכיון הנתון

int check(char board[][X], char str[], int length, int place\_raw, int place\_col, int go\_raw, int go\_col)//בדיקת כיוון אחד

{

if (place\_raw + (go\_raw \* length) >= 0 && place\_col + (go\_col \* length) >= 0)//אם לא יוצא מהגבולות

{

char str2[X];

int i;

for (i = 0; i < length; i++)

str2[i] = board[place\_raw + (go\_raw \* i)][place\_col + (go\_col \* i)];

str2[length] = 0;

if (!strcmp(str, str2))

return 1;

}

return -1;

}

//פונקציה הבודקת את כל הכיוונים ומפעילה בכל כיוון את צ'ק

int looking\_around(char board[][X], char str[], int length, int place\_raw, int place\_col)

{

int flag = -1;

int raw = 0, col = 0;

for (int a = -1; a < 2; a++)

{

for (int b = -1; b < 2; b++)

{

flag = check(board, str, length, place\_raw, place\_col, a, b);

if (flag == 1)

break;

}

if (flag == 1)

break;

}

return flag;

}

//פונקציה אשר בודקת האם בתוך הלוח קיים הסטרינג הנוכחי

int where\_there\_the\_word\_is(char board[][X], char str[], int length, int\* place\_raw, int\* place\_col)

{

int pc, pr;

int flag;

for (pr = 0; pr < X; pr++)

{

for (pc = 0; pc < X; pc++)

{

if (str[0] == board[pr][pc])

{

flag = looking\_around(board, str, length, pr, pc);

if (flag == 1)

{

\*place\_col = pc;

\*place\_raw = pr;

return flag;

}

}

}

}

return flag;

}

void main()

{

srand(time(NULL));

int x;

int place\_raw = 0;

int place\_col = 0;

char board[X][X];

printf("press 1 to fill the board randomaly\npress 2 to fill the board by yourself\t");

scanf("%d", &x);

if (x == 1)

fill\_board\_randomally(board);

else if (x == 2)

fill\_board\_by\_yourself(board);

print\_board(board);

printf("please enter the word to search\n");

char str[X] = "";

scanf("%s", str);

int length = strlen(str);

x = where\_there\_the\_word\_is(board, str, length, &place\_raw, &place\_col);

if (x == -1)

printf("Not Found");

if (x == 1)

printf("pos: [%d][%d]", place\_raw, place\_col);

}

שאלה מספר 3

השלמות

סעיף א'

1. 24
2. sizeof(pupilPtr)
3. 3
4. classes[8\*k]

סעיף ב'

1. buff->level
2. buff->clas
3. school[row][col]
4. q=p
5. q->next
6. q

סעיף ג'

1. fopen(“Grades.dat”,”rb”)
2. fread(buffin,sizeof(buffer),1,fp)
3. insertGrade(buff)
4. fread(buffin,sizeof(buffer),1,fp)

סעיף ד'

1. p->grade>=90
2. (pupilPtr)realloc(pupils,sizeof(pupilPtr)\*(n+1))
3. pupils[n].grade
4. p=p->next
5. sizeof(pupil)
6. comp

סעיף ה'

1. school[level][0]
2. q->grade>pmax->grade
3. q

סעיף ו'

1. p->grade
2. freq[k]
3. p=p->next
4. max

סעיף ז'

1. “wb”
2. fwrite(pupils,sizeof(pupil),fp)

סעיף ח'

1. frequentGrade()
2. dilligentPupils(level)
3. (student)bsearch(&st, pupils, sizeof(pupils)/sizeof(Pupil), sizeof(Pupil), comp)
4. electedPupil(level)