Правительство Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего профессионального образования «Национальный исследовательский университет «Высшая школа экономики»

Московский институт электроники и математики Национального исследовательского университета «Высшая школа экономики»

Факультет прикладной математики и кибернетики

О Т Ч Е Т

По лабораторной работе № 12

По курсу «Программирование»

ФИО студента	Номер группы	Дата	Баллы
Борисов Павел Геннадьевич	ПИ-11		

Москва - 2014 г.

Постановка задания

Задать многочлен от X односвязным списком. Элемент списка содержит неотрицательный целочисленный показатель степени X и ненулевой коэффициент при этой степени (в списке не должно быть элементов с одинаковыми степенями). Составить программу, включающую помимо указанных в задании функций, функции создания и вывода списка на экран. Список или списки должны отображаться на экране до обработки и после.

Вариант 3

Написать функцию удаления коэффициента из представления многочлена (всех элементов, имеющих заданный коэффициент при разных степенях)

Makefile

```
lab12: main.c linkedlist.c linkedlist.h
	gcc --std=c99 -ggdb -o lab12 main.c linkedlist.c linkedlist.h

clean:
	rm -f lab12
```

linkedlist.c

```
#include "linkedlist.h"
#include <string.h>

void printPolynomial(Polynomial* pol)
{
  static int nosign = 1;
  if (!pol)
    {
      printf("\n");
      nosign = 1;
      return;
    }
  char format[10];
  memset(format, '\0', 10);
  if (pol->coeff == 1)
    {
      if (!nosign)
        strcat(format, " + ");
      if (pol->degree == 0)
        strcat(format, "1");
    }
  else if (pol->coeff == -1)
    {
      strcat(format, " - ");
      if (pol->degree == 0)
        strcat(format, "1");
    }
  else
    {
```

```
30        if (nosign)
31          strcat(format, "%d");
32        else
33          strcat(format, " %+d");
34      }
35    if (pol->degree == 1)
36      strcat(format, "x");
37    else if (pol->degree > 1)
38      strcat(format, "x^%d");
39
40    switch(pol->coeff)
41      {
42      case 1: case -1:
43        printf(format, pol->degree);
44        break;
45      default:
46        printf(format, pol->coeff, pol->degree);
47      }
48
49    nosign = 0;
50    printPolynomial(pol->next);
51  }
52
53  Polynomial* createPolynomial(unsigned int *coeffs, int coeffsCount)
54  {
55    Polynomial* pol = malloc(sizeof(struct linked_list));
56    int input;
57    printf("Enter the degree of polynomial (negative to stop): ");
58
59    if ((!scanf("%d", &input)) || input < 0)
60      {
61        free(pol);
62        if (coeffs)
63          free(coeffs);
64        return NULL;
65      }
66    if (!coeffs)
67      {
68        coeffs = calloc(1, sizeof(int));
69        coeffsCount = 1;
70      }
71    else
72      {
73        for (int i = 0; i < coeffsCount; ++i)
74          if (coeffs[i] == input)
75            {
76              printf("This degree has already entered. Enter another degree\n");
77              free(pol);
78              return createPolynomial(coeffs, coeffsCount);
79            }
80        coeffs = realloc(coeffs,++coeffsCount*sizeof(int));
81      }
82
83    coeffs[coeffsCount-1] = input;
84    pol->degree = input;
85    printf("Enter the coefficient[%d] (0 to stop):", pol->degree);
```

```
86
87    if ((!scanf("%d", &input)) || input == 0)
88      {
89        free(pol);
90        if (coeffs)
91          free(coeffs);
92        return NULL;
93      }
94    pol->coeff = input;
95    pol->next = createPolynomial(NULL, 0);
96    return pol;
97  }
98
99  void cleanPolynomial(Polynomial* pol)
100 {
101   Polynomial* p_next = pol->next;
102   free(pol);
103   if (p_next)
104     cleanPolynomial(p_next);
105 }
106
107
108 Polynomial* removeCoefficient(int coefficient, Polynomial* pol)
109 {
110   while(pol) //set start of the new list
111     {
112       if (pol->coeff != coefficient)
113         break;
114       else
115         {
116           Polynomial* p_next = pol->next;
117           free(pol);
118           pol = p_next;
119         }
120     }
121   if (!pol)
122     return NULL;
123
124   Polynomial* p = pol;
125   Polynomial* p_prev;
126   while(p_prev = p, p = p->next)
127     {
128       if(p->coeff == coefficient)
129         {
130           Polynomial* p_next = p->next;
131           p_prev->next = p_next;
132           free(p);
133           p = p_prev;
134         }
135     }
136   return pol;
137 }
```

linkedlist.h

0.1. main.c

```
/* Moscow Institute of Electronics and Mathematics
   The Faculty of Applied Mathematics

   Assignment #N
   Language: C99
   Compiler: gcc

   Student: Pavel Borisov
   Group: Applied Informatics 11
*/

#include "linkedlist.h"

int main()
{
  Polynomial *p = createPolynomial(NULL, 0);
  if (!p)
    {
      printf("No polynomial entered. Exiting\n");
      return -1;
    }
  printf("Input: ");
  printPolynomial(p);
  printf("Enter the coefficient for removing: ");
  int coeff;
  scanf("%d", &coeff);
  p = removeCoefficient(coeff, p);
  printf("Result: ");
  printPolynomial(p);
  return 0;
}
```

Тесты

```
pasha@primum ~/projects/miem/1/4/12 (git)-[master] % ./lab12
Enter the degree of polynomial (negative to stop): 3
Enter the coefficient[3] (0 to stop):4
Enter the degree of polynomial (negative to stop): 2
Enter the coefficient[2] (0 to stop):4
Enter the degree of polynomial (negative to stop): 1
Enter the coefficient[1] (0 to stop):-5
Enter the degree of polynomial (negative to stop): 0
Enter the coefficient[0] (0 to stop):1
Enter the degree of polynomial (negative to stop): -1
Input: 4x^3 +4x^2 -5x + 1
Enter the coefficient for removing: 4
Result: -5x + 1
```

```
 ./lab12
Enter the degree of polynomial (negative to stop): 5
Enter the coefficient[5] (0 to stop):1
Enter the degree of polynomial (negative to stop): 3
Enter the coefficient[3] (0 to stop):2
Enter the degree of polynomial (negative to stop): 6
Enter the coefficient[6] (0 to stop):-2
Enter the degree of polynomial (negative to stop): -1
Input: x^5 +2x^3 -2x^6
Enter the coefficient for removing: 2
Result: x^5 -2x^6
```