С++

**Задача:**

 Задан текст, состоящий из «0» и «1». Оставив первый символ в неприкосновенности, заменить каждый следующий символ на «0», если предыдущий не совпадает с рассматриваемым, и «1» – в противном случае.

**Пример как нужно выполнить**

10.4. ДИНАМИЧЕСКОЕ РАСПРЕДЕЛЕНИЕ ПАМЯТИ, СПИСКИ

Цель работы

Лабораторная работа предназначена для изучения способов представления и обработки

списков в языке С. В работе следует написать функцию, выполняющую одну из операций

над списком. Информационная часть элемента списка может быть любой.

Пример выполнения работы

Написать функцию, которая включает новый узел в голову списка.

#include <stdio.h>

#include <conio.h>

#include <alloc.h>

typedef struct EList{

 char data;

 struct EList \*next;

 } EList;

EList\* AddElem( EList\*, char ); /\* Включение элемента \*/

void PrnList( EList\* ); /\* Печать списка \*/

void main(){

EList\* Head=NULL; /\* Указатель на голову списка \*/

EList\* t;

char c;

/\* Ввод данных и формирование списка \*/

printf("\nВведите символы-элементы списка (завершение ввода-пробел).");

printf("\nСимволы вводить без Enter после каждого символа");

printf("\nВводите;");

while( (c = getch()) != ' ' ){

 putch( c );

 Head = AddElem( Head, c );

 }

/\* Вывод результатов \*/

printf( "\nРезультат:" );

PrnList( Head );

/\* Освобождение памяти, занятой списком \*/

while( Head != NULL ){

 t = Head;

 Head = Head->next;

 free( t );

 }

}/\* main \*/

EList \*AddElem( EList \*h, char inf )

/\*\*\*\*\*\*\*\*\*\*\* AddElem \*\*\*\*\*\*\*\*\*\*

Действие:

 Включает элемент в голову списка.

Параметры:

 h - указатель на голову списка, – 36 –

© 1998 Калачев Д.П., Лутай В.Н.

 inf - информационная часть добавляемого элемента.

Возвращает:

 указатель на голову списка или

 NULL, если нет памяти для нового элемента.

Замечания: -

\*/

{

EList \*buf;

if( !( buf = (EList \*)malloc( sizeof( EList ) )))

 return( NULL );

buf->data = inf;

buf->next = h;

return( buf );

}/\* AddElem \*/

void PrnList( EList \*h )

/\*\*\*\*\*\*\*\*\*\*\* PrnList \*\*\*\*\*\*\*\*\*\*

Действие:

 Печатает список.

Параметры:

 h - указатель на голову списка,

Возвращает: -

Замечания: -

\*/

{

printf( "\nСписок имеет следующий вид: " );

while( h != NULL ){

 printf( "%c", h->data );

 h = h->next;

 }

}/\* PrnList \*/

Задача: Задан текст, состоящий из «0» и «1». Оставив первый символ в неприкосновенности, заменить каждый следующий символ на «0», если предыдущий не совпадает с рассматриваемым, и «1» – в противном случае.