**РУСЕНСКИ УНИВЕРСИТЕТ “АНГЕЛ КЪНЧЕВ”**

**КУРСОВА РАБОТА**

по

САА

Студент:

Факултетен номер:

Група:

Специалност:

Дата: ПРОВЕРИЛ:.....................

РУСЕ

2014 г.

# Задача.

17. По време на Втората световна война е била измислена машина, кодираща съобщенията по следния начин. След съставяне на съобщението, то бива разбито на части от по два символа. Между всеки две последователни части се поставя кодова дума – „пе“. Така полученото съобщение бива изпратено и съответно разкодирано от съюзниците.

Пример:

„Целият ни взвод е пленен.“ –>

„Цепелипеятпе нпеи певзпевопед пее пеплпеенпеенпе.“

Да се разработи програма, която кодира и разкодира съобщения за целите на армията. Програма да има следните подпрограми:

* Подпрограма разделяща изречението (запазено като низ) на съставни части от по два символа (ако Ви затруднява, въвеждайте изречението на съставни части от по два символа от клавиатурата). Тези части трябва да се запазят в списък.
* Подпрограма за кодиране на съобщението (добавяне на кодовата дума между всеки две от съседните части).
* [Подпрограма за изтриване на кодовите думи от списъка.]
* Подпрограма за извеждане на елементите от списъка.

# Описание на решението.

Разделянето на изречението на съставни части от по два символа се прави в главната програма main(). Всяка част се добавя в края на списъка чрез функцията Add(), т.е списъкът е от тип опашка (FIFO).

Кодирането на изречението става чрез Encode(). Кодовата дума се добавя в списъка, използвайки схемата „вмъкване след елемент, сочен от даден указател P != NULL“

**Постъпково изпълнение на кода:**

1. След изпълнение на оператора

new(Pnew);

състоянието на списъка е:

**Pnew**

**…**

**…**

**5**

**6**

**12**

**P**

1. След изпълнение на оператора

readln(Pnew^.info);

състоянието на списъка е:

**7**

**Pnew**

**…**

**…**

**5**

**6**

**12**

**P**

1. След изпълнение на оператора

Pnew^.next:=P^.next;

състоянието на списъка е:

**7**

**Pnew**

**…**

**…**

**5**

**6**

**12**

**P**

1. След изпълнение на оператора

P^.next:=Pnew;

състоянието на списъка е:

**7**

**Pnew**

**…**

**…**

**5**

**6**

**12**

**P**

Декодирането на изречението става чрез функцията Decode(). Като параметър не се подава кодовата дума, т.к позициите на кодовата дума в изречението може да се „изчисли“. Използва се схема „Изтриване след елемент, сочен от даден указател P“.

**Постъпково изпълнение на кода:**

1. След изпълнение на оператора

Pdel:=P^.next;

състоянието на списъка е:

**…**

**12**

**5**

**6**

**7**

**…**

**P**

**Pdel**

1. След изпълнение на оператора

P^.next:= P^.next^.next;

състоянието на списъка е:

**…**

**12**

**5**

**6**

**7**

**…**

**P**

**Pdel**

1. След изпълнение на оператора

Dispose(Pdel);

състоянието на списъка е:

**…**

**12**

**5**

**6**

**…**

**P**

# Описание на използваните функции.

**(1)**

Вид на функцията: **int main();**

Действие: **главна програма (функция);**

Параметри: **няма;**

Използвани глобални променливи: **няма;**

Извиквани функции: **Add(), Print(), Encode(), Decode();**

**(2)**

Вид на функцията: **void Add(PMessage\* head\_ref, string msg)**

Действие: **подпрограма за добавяне на нов елемент в списъка;**

Параметри: **PMessage\* head\_ref** – указател към началото на списъка;

**string msg** – думата за добавяне;

Използвани глобални променливи: **няма;**

Извиквани функции: **няма***;*

**(3)**

Вид на функцията: **void Print(PMessage head\_ref)**

Действие: **подпрограма за извеждане на списъка;**

Параметри: **PMessage head\_ref** *– указател към началото на списъка;*

Използвани глобални променливи: **няма;**

Извиквани функции: **няма**;

**(4)**

Вид на функцията: **void Encode(PMessage &head\_ref, string code)**

Действие: **подпрограма за кодиране на изречение;**

Параметри: **PMessage &head\_ref** *– указател към началото на списъка;*

**string code** – кодова дума;

Използвани глобални променливи: **няма;**

Извиквани функции: **няма*;***

**(5)**

Вид на функцията: **void Decode(PMessage &head\_ref)**

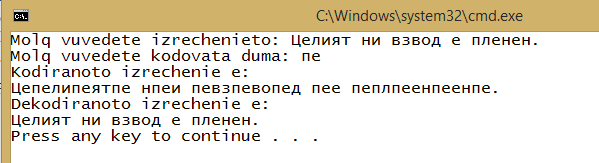
Действие: **подпрограма за декодиране на изречение;**

Параметри: **PMessage &head\_ref** *– указател към началото на списъка;*

Използвани глобални променливи: **няма;**

Извиквани функции:**няма***;*

# Тестов пример.



# Код на програмата.

#include <iostream>

#include <string>

using namespace std;

typedef struct Message

{

char msg[2];

Message \*next;

} \*PMessage;

void Add(PMessage\* head\_ref, string msg)

{

PMessage current, new\_node;

new\_node = new Message;

strcpy(new\_node->msg, msg.c\_str());

if (\*head\_ref == NULL)

{

new\_node->next = \*head\_ref;

\*head\_ref = new\_node;

}

else

{

current = \*head\_ref;

while (current->next != NULL)

current = current->next;

new\_node->next = current->next;

current->next = new\_node;

}

}

void Print(PMessage head\_ref)

{

while (head\_ref)

{

cout << head\_ref->msg;

head\_ref = head\_ref->next;

}

}

void Encode(PMessage &head\_ref, string code)

{

PMessage new\_item, p = head\_ref;

while (p && strlen(p->msg) > 1)

{

new\_item = new Message;

strcpy(new\_item->msg, code.c\_str());

new\_item->next = p->next;

p->next = new\_item;

p = p->next->next;

}

}

void Decode(PMessage &head\_ref, string code)

{

PMessage p = head\_ref, pdel;

while (p && strlen(p->msg) > 1)

{

pdel = p->next;

p->next = p->next->next;

delete pdel;

p = p->next;

}

}

int main()

{

PMessage head = NULL;

string fullMsg, code;

cout << "Molq vuvedete izrechenieto: ";

getline(cin, fullMsg);

for (int i = 0; i < fullMsg.length(); i += 2)

{

Add(&head, fullMsg.substr(i, 2));

}

cout << "Molq vuvedete kodovata duma: ";

cin >> code;

cout << "Kodiranoto izrechenie e:" << endl;

Encode(head, code);

Print(head);

cout << endl << "Dekodiranoto izrechenie e:" << endl;

Decode(head);

Print(head);

cout << endl;

return 0;

}