Программирование на языках высокого уровня ч2

**Лабораторная №1**

**Задача 1**

Составить программу на языке С/С++, содержащую:

- объявления и инициализацию указателей на различные типы объектов:

переменные, массивы, структуры;

- объявления и инициализацию указателей с различными модификаторами (const, near, far, huge);

- преобразования указателей различных типов, в том числе типа void.

#include <stdio.h>

#include <conio.h>

#include <iostream.h>

void main(void)

{

//переменные

const double Eu=8.1;

double \*pEu=(double\*)&Eu;

\*pEu=4.6;

//изменения константы Eu с помощью указателя

cout<<Eu<<"\n";

//массивы

float Y[]={1,2,3,4,5,6}; //указатель на масив вещественного типа

float \*pY=Y; //pY указывает на начало массива y[]

//структуры

struct komp

{

char proc [20];

int memory;

};

komp \*Celeron=NULL; //инициализированный указатель на объект типа komp

komp MStar[]={"Celeron2000",512,"Pentium2500",512};

Celeron=&MStar[0];

cout<<"\n"<<Celeron<<"\t"<<Celeron->proc<<"\t"<<Celeron->memory;

//модификаторами (const, near, far, huge)

double St[]={8.1,3.3,1.4};

double \*const pSt=&St[2]; //инициализированный константный указатель

//на элемент массива типа float

//преобразования указателей различных типов

unsigned long A=12345678;

char \*char\_=(char\*)&A; //указатель на объект типа char

int \*int\_=(int\*)&A; //указатель на объект целого типа

float \*float\_=(float\*)&A; //указатель на объект вещественного типа

cout<<A<<"\t"<<(void\*)char\_<<"\t"<<int\_<<"\t"<<float\_<<"\n";

cout<<A<<"\t"<<\*char\_<<"\t"<<\*int\_<<"\t"<<\*float\_;

cerr<<" ";

void \*vp;

int i=77;

float f=2.7;

vp=&i;

cout<<"\*(int\*)vp="<<\*(int\*)vp;

vp=&f;

cout<<"\*(float\*)vp="<<\*(float\*)vp;

cerr<<" ";

getch();

}

**Задача 2**

Составить программу на языке С/С++, содержащую:

- простейшие действия с одномерными массивами (как числовыми, так и символьными),

используя указатели, методы доступа к элементам массивов через указатели;

- простейшие действия с многомерными массивами (как числовыми, так и символьными),

используя указатели, методы доступа к элементам массивов через указатели;

- всевозможные операции с указателями.

\*/

#include <stdio.h>

#include <conio.h>

#include <iostream.h>

void main(void)

{

//одномерные массивы

int x[]={4,5,6,7,2,4,6};

char ch[]="ABCD";

char ch1[]={'A','B','C','D'};

int \*px=x;

char \*pch=ch;

char \*pch1=ch1;

for(int i=0;i<7;i++) cout<<\*(px+i)<<"\t";

cout<<pch<<"\n";

for(i=3;i>=0;i--) cout<<\*(pch1+i)<<"\t";

cout<<"\n";

for(i=0;i<4;i++) cout<<\*(pch1++)<<"\t";

//многомерные массивы

int A[3][3]={1,2,3,4,5,6,7,8,9};

char B[3][3]={'A','B','C','D','E','F','G','H','\0'};

cout<<\*(\*(A+1))<<"\n";

int C[3][2][4]=

{ 0, 1, 2, 3,

10, 11, 12, 13,

100, 101, 102, 103,

110, 111, 112, 113,

200, 201, 202, 203,

210, 211, 212, 213

};

cout<<\*\*\*C<<"\n";

cout<<\*(\*(\*(C+1)+1)+3)<<"\n";

int sum=0;

for(int i=0;i<3;i++)

for (int j=0;j<2;j++)

{

cout<<"\n";

for(int k=0;k<4;k++)

{

sum+=\*(\*(\*(C+i)+j)+k);

cout<<\*(\*(\*(C+i)+j)+k)<<"\t";

}

}

cout<<"\n"<<sum;

cerr<<" ";

}

**Задача 3**

Составить программу на языке С/С++, содержащую:

- объявления указателей на функции и передачи аргументов с помощью указателей;

- доступ к содержимым по указателям.

\*/

#include <stdio.h>

#include <conio.h>

#include <iostream.h>

void f1(int \*x)

{

cout<<\*x;

}

int\*\* f2(int \*x)

{

int \*\*z= new int\*[\*x];

for (int i=0;i<\*x;i++) z[i] = new int[\*x];

for (i=0;i<(\*x);i++)

for (int j=0;j<(\*x);j++)

\*(\*(z+i)+j)=i+j;

return z;

}

void main(void)

{

int A=5;

void (\*fun)(int\*);

fun=f1;

int \*pa=&A;

(\*fun)(pa);

int \*\*pf=NULL;

int (\*\*(\*fun1)(int\*));

fun1=f2;

pf=(\*f2)(pa);

cout<<"\n"<<\*\*pf;

for (int i=0;i<(\*pa);i++)

{

cout<<"\n";

for (int j=0;j<(\*pa);j++)

cout<<\*(\*(pf+i)+j)<<"\t";

}

cerr<<" ";

}

**Задача 4**

Составить программу на языке С/С++, содержащую:

- динамическое размещение массивов;

- интерпретацию составных описателей.

**11**

**Задача 1**

а) Создать программу определения факториала числа n, использующую цикл (любой - for или while).

б) Составить описание рекурсия математическими терминами.

в) Разработать схему рекурсивных вызовов.

г) Составить программу определения факториала числа n, использующую рекурсию.

д) Проанализировать работу рекурсивной функции с помощью отладчика.

е) Сделать сравнительный анализ работы двух программ.

\*/

а)

#include <iostream.h>

void main()

{

int i;

cout << "Введите число:";

cin >> i;

// расчет факториала

int j, fact=1;

for (j=i; j>=1; j--) fact\*=j;

cout << "Факториал равен:" << fact;

}

г)

#include <iostream>

long fact(int f)

{

if (f<0) return (0);

if (f==0) return (1);

return (f\*fact(f-1));

}

**Задача 2**

Написать программу рекурсивной функции Аккермана с использованием основной функции ackr и

вспомогательной функции smacc. Составить описание рекурсия математическими

терминами. Разработать схему рекурсивных вызовов. Проанализировать работу рекурсивной

функция с помощью отладчика.

\*/

#include <stdio.h>

#include <conio.h>

void main(void)

{

//clrscr();

int x,y,n;

long t;

int ackr(int,int,int);

printf("\nВведите 3 целых положительных числа:\n");

n=3,x=3,y=3;

//scanf("%d%d%d",&n,%x,&y);

t=ackr(n,x,y);

printf("Результат вычисления функции Акермана = %ld",t);

getch();

}

int smacc (int n, int x)

{

switch (n)

{

case 0: return (x+1);

case 1: return (x);

case 2: return (0);

case 3: return (1);

default: return (2);

}

}

int ackr(int n, int x, int y)

{

int z;

int smacc(int,int);

if(n==0||y==0)z=smacc(n,x);

else

{

z=ackr(n,x,y-1);

z=ackr(n-1,z,x);

}

return z;

}

**Задача З**

Подобрать собственный пример, иллюстрирующий работу рекурсивной функции.

Составить программу, выполнить ее анализ.

\*/

Рекурсивный ввод и печать списка.

#include <stdlib.h>

#include <stdio.h>

struct cell

{

char sign[10];

int weight;

struct cell \*pc;

};

struct cell\* input(void)

{

struct cell \*p;

p=(struct cell \*)malloc(sizeof(struct cell));

printf("Sign=");

scanf("%s",& p->sign);

printf("Weight=");

scanf("%d",& p->weight);

if(p->weight==0)

{

free(p);

return NULL;

}

p->pc=input();

return p;

}

void output(struct cell \*p)

{

if (p==NULL)

{

printf("\nEND");

return;

}

printf("\nsign=%s\tweight=%d",p->sign,p->weight);

output(p->pc);

}

void main(void)

{

struct cell \*beg=NULL;

beg=input();

printf("\nBegin");

output(beg);

}

**12**

**Задача 1**

Создать несвязанный числовой список в виде массива. Распечатать его.

Выполнить следующие операции с несвязанным списком:

а) найти максимальный по модулю элемент;

б) вставить за ним новый элемент, равный по величине, но противоположный по знаку;

в) Новый список распечатать.

\*/

#include <stdio.h>

#include <conio.h>

#include <iostream.h>

int abc (int x) {return x>0?x:-x;}

void main(void)

{

int d[100]={1,2,300,-4,-50,6,7},L=7;

int max = 0;

//максимальный по модулю элемент

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

if(abc(d[i])>abc(d[max])) max=i;

printf("MAX element= |%d|",d[max]);

//добавление нового элемента за МАХ элеме

for (int j=L-1; j>max; j--)

d[j+1]=d[j];

d[max+1]=-d[max]; L++;

//исключить первый элемент списка

for (j=0; j<L; j++)

d[j]=d[j+1];

L--;

//новый список

cout<<"\n";

for (i=0;i<L;i++) cout<<d[i]<<"\t";

cerr<<" ";

}

**Задача 2**

Создать связанный числовой список. Распечатать его. Выполнить следующие операции со связанным списком:

а) найти максимальный по модулю элемент;

б) вставить за ним новый элемент, равный по величине, но противоположный по знаку;

в) исключить первый элемент списка. Новый список распечатать.

\*/

#include <stdio.h>

#include <conio.h>

#include <iostream.h>

int abc (int x) {return x>0?x:-x;}

struct ND

{

int val;

struct ND \* n;

} \*dl=NULL, \*r, \*p=NULL;

void main(void)

{

int mas[]={1,2,3,-50,6,30,1,2,3};

for(int i=0;i<sizeof(mas)/sizeof(mas[0]);i++)

{

r=new ND;

r->val=mas[i];

r->n=NULL;

if(dl==NULL)

{

dl=r;

p=r;

}

else

{

p->n=r;

p=r;

}

}

//максимальный по модулю элемент

p=dl;

int MAX=dl->val;

while((p->n)!=NULL)

{

p=p->n;

if(abc(p->val)>abc(MAX)) MAX=p->val;

}

cout<<"\n";

printf("MAX element= |%d|",MAX);

//добавление нового элемента за МАХ

p=dl;

while((p->n)!=NULL)

{

p=p->n;

if(p->val==MAX)

{

r=new ND;

r->n=p->n;

r->val=-MAX;

p->n=r;

}

}

//исключить первый элемент списка

p=dl;

dl=dl->n;

delete (p);

//новый список

p=dl;

cout<<"\nspisok\n";

while(p!=NULL)

{

cout<<p->val<<"\t";

p=p->n;

}

cerr<<" ";

}

**Задача 3**

Создать связанный числовой список типа очередь из 5 объектов. Распечатать его. Выполнить следующие операции со связанным списком:

а) добавить два новых объекта в очередь. Распечатать очередь;

б) поменять местами первый и последний объект в очереди. Распечатать очередь;

в) удалить из очереди первых три объекта. Распечатать очередь.

\*/

#include <stdio.h>

#include <conio.h>

#include <iostream.h>

#include <stdlib.h>

struct elem {

int inf;

struct elem \*link;

} \*begq=NULL, \*endq=NULL;

void addel(int val)

{

struct elem \*p= new struct elem;

p->inf=val;

p->link=NULL;

if(endq==NULL) begq=p;

else endq->link=p;

endq=p;

}

int getdelel(void)

{

struct elem \*p;

int temp;

temp=begq->inf;

p=begq;

begq=p->link;

if(begq==NULL) endq=NULL;

delete p;

return temp;

}

void output(struct elem \*p)

{

if (p==NULL)

{

printf("\nEND");

return;

}

printf("\tinf=%i",p->inf);

output(p->link);

}

void main(void)

{

int mas[]={10,20,30,1,2,3};

for (int i=0;i<sizeof(mas)/sizeof(mas[0]);i++) addel(mas[i]);

output(begq);

int number;

while (1)

{

printf("\n 1 - добавить два новых объекта в очередь");

printf("\n 2 - поменять местами первый и последний объект в очереди");

printf("\n 3 - удалить из очереди первых три объекта");

printf("\n 4 - Exit");

while (1)

{

printf("\n Menu:");

scanf("%d",&number);

if(number>0&&number<5) break;

printf("\n Error menu");

while (getchar()!='\n');

}

switch (number)

{

case 1:

{

//а) добавить два новых объекта в очередь. Распечатать очередь;

scanf("%d",&number);

addel(number);

scanf("%d",&number);

addel(number);

break;

}

case 2:

{

//б) поменять местами первый и последний объект в очереди. Распечатать очередь;

int temp = getdelel();

int endelement= endq->inf;

while (begq->inf != endelement) addel(getdelel());

addel(temp);

break;

}

case 3:

{

//в) удалить из очереди первых три объекта. Распечатать очередь.

for (i=0;i<2;i++) getdelel();

break;

}

case 4: exit(0); break;

default: exit(0);

}

printf("\nnew spisok\n");

output(begq);

cerr<<" ";

getch();

}

}

/\*

struct cell\* input(void)

{

struct cell \*p;

p=(struct cell \*)malloc(sizeof(struct cell));

printf("Sign=");

scanf("%s",& p->sign);

printf("Weight=");

scanf("%d",& p->weight);

if(p->weight==0)

{

free(p);

return NULL;

}

p->pc=input();

return p;

}

**Задача 4**

Создать связанный числовой список типа стек из 5 объектов. Распечатать его. Выполнить следующие операции со связанным списком:

а) добавить два новых объекта в стек. Распечатать стек;

б) поменять местами первый и последний объект в стека. Распечатать стек;

в) удалить из стека первых три объекта. Распечатать стек.

**15**

/\*

struct list

{

int value;

struct list \*next;

};

list \*hear=NULL;

int count=0;

//void show

void show(struct cell \*p)

{

if (p==NULL)

{

printf("\nEND");

return;

}

printf("\nadress=%p\tvalue=%d",p,p->value);

output(p->pc);

}

void add\_head (long value)

{

count++;

list \*old\_head=head;

head=(struct list \*)malloc(sizeof(struct list));

head->next=old\_old\_head;

head->value=value;

}

void insert (int pos,long value)

{

list \*target;

list \*old\_next;

int i=0;

if(head!=NULL)

{

target=head;

while((i<pos)&&(target!=NULL))

{

target=target->next;

i++;

}

if(i!=pos) return;

old\_next=target->next;

target=target->next=(struct list \*)malloc(sizeof(struct list));

target->value=value;

target->next=old\_next;

count+;

}

else

{

target=head=(struct list \*)malloc(sizeof(struct list));

target->next=NULL;

target->value=value;

count+;

}

}

void delete\_any (int pos)

{

if(pos<0) return;

list \*previous;

int i=0;

if(head!=NULL)

{

if(pos==0)

{

list \*next\_item=head->next;

free(head);

head=next\_item;

}

else

{

previous=head;

pos--;

while((i<pos)&&(previous!=NULL))

{

previous=previous->next;

i++;

}

if(i!=pos) return;

list \*next\_item=previous->next->next;

free(previous->next);

previous->next=next\_item;

}

}

\*/

**16**