11.1

1.Постановка задачи

Создать некоторую структуру с указателем на некоторую функцию в качестве поля. Вызвать эту функцию через имя переменной этой структуры и поле указателя на функцию.

2.Код программы

#include <stdio.h>

#include <stdlib.h>

struct s

{

void (\*print)();

};

void print(){

puts("1");

}

int main()

{

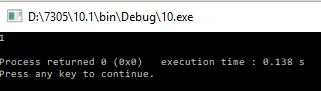
struct s funct={&print};

funct.print();

return 0;

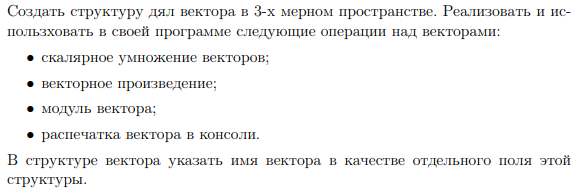
}

3.Результат



11.2

1.Постановка задачи



2.Список идентификаторов

|  |  |  |
| --- | --- | --- |
| Имя | Смысл | Тип |
| x | Координата x, поле структуры | float |
| y | Координата y, поле структуры | float |
| z | Координата z, поле структуры | float |
| name | Имя вектора | char |

3.Код программы

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

struct vector

{

char name;

float x;

float y;

float z;

};

void print(struct vector v)

{

printf("%f\n%f\n%f",v.x,v.y,v.z);

}

float modul(struct vector v)

{

v.x=v.x\*v.x;

v.y=v.y\*v.y;

v.z=v.z\*v.z;

float p=sqrt(v.x+v.z+v.y);

return p;

};

struct vector vpr(struct vector a,struct vector b)

{

struct vector v;

v.x=a.y\*b.z-b.y\*a.z;

v.y=-(a.x\*b.z-b.x\*a.z);

v.z=a.x\*b.y-b.x\*a.y;

return v;

}

float cs(struct vector a, struct vector b)

{

struct vector v;

float al=(a.x\*b.x+a.y\*b.y)/sqrt(a.x\*a.x+a.y\*a.y)\*sqrt(b.x\*b.x+b.y\*b.y);

return al;

}

int main()

{

float scp;

struct vector a;

a.name='a';

puts("Enter coordinates of the a");

scanf("%f",&a.x);

scanf("%f",&a.y);

scanf("%f",&a.z);

struct vector b;

b.name='b';

puts("Enter coordinates of the b");

scanf("%f",&b.x);

scanf("%f",&b.y);

scanf("%f",&b.z);

printf("a=%f\nb=%f",modul(a),modul(b));

puts("\na\*b");

print(vpr(a,b));

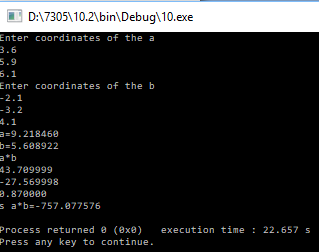
scp=modul(a)\*modul(b)\*cs(a,b);

printf("\ns a\*b=%f\n",scp);

return 0;

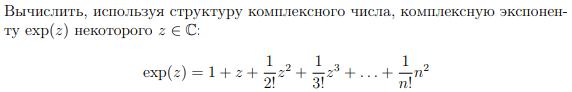
}

4.Результат



11.3

1.Постановка задачи



2.Математическая модель



3.Список идентификаторов

|  |  |  |
| --- | --- | --- |
| Имя | Смысл | Тип |
| x | Вещественная часть комплексного числа, поле структуры | float |
| y | Коэффициент при мнимой части комплексного числа, поле структуры | float |
| ph | Угол между векторами | float |
| i | Параметр цикла | int |
| p | Факториал | float |
| n | Число слагаемых | int |

4.Код программы

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

struct c

{

float x;

float y;

};

int main()

{

int n;

struct c z;

puts("Enter a and b\n");

scanf("%f",&z.x);

scanf("%f",&z.y);

puts("Enter n");

scanf("%d",&n);

float ph=z.y/z.x;

z.x=1;

z.y=1;

for(int i=1, p=1;i<n;i++)

{

if(i!=0)

p=p\*i;

z.x+=1/p\*cos(i\*ph);

z.y+=1/p\*sin(i\*ph);

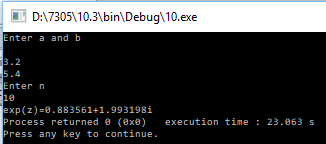
}

printf("exp(z)=%f+%fi",z.x,z.y);

return 0;

}

5.Результат



11.4

1.Постановка задачи

Реализовать в виде структур двунаправленный связный список и совершить отдельно его обход в прямом и обратном направлениях с распечаткой значений каждого элемента списка.

2.Список идентификаторов

|  |  |  |
| --- | --- | --- |
| Имя | Смысл | Тип |
| d | Число, поле структуры | int |
| \*next | Указатель на следующий элемент | \*struct |
| \*prev | Указатель на предыдущий элемент | \*struct |
| i | Параметр цикла | int |
| numb | Число | int |

3.Код программы

#include <stdio.h>

#include <stdlib.h>

struct uz

{

int d;

struct uz \*next;

struct uz \*prev;

};

struct uz \*init(int a)

{

struct uz \*list;

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

list->d=a;

list->next=NULL;

list->prev=NULL;

return list;

}

struct uz \*addel(struct uz \*list, int n)

{

struct uz \*temp, \*p;

temp=(struct uz\*)malloc(sizeof(struct uz));

p=list->next;

list->next=temp;

temp->d=n;

temp->next=p;

temp->prev=list;

if(p!=NULL)

p->prev=temp;

return temp;

}

void print(struct uz \*list)

{

struct uz \*p;

p=list;

do

{

printf("%d ",p->d);

p=p->next;

} while (p!=NULL);

}

void obrprint(struct uz \*list)

{

struct uz \*p;

p=list;

while (p->next!=NULL)

p = p->next;

do {

printf("%d ",p->d);

p=p->prev;

} while (p!=NULL);

}

int main()

int numb;

{

struct uz \*head, \*currn;

head=init(4);

currn=head;

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

{

scanf("%d",&numb);

currn=addel(currn,numb);

}

print(head);

printf("\n");

obrprint(head);

return 0;

}

4.Результат

