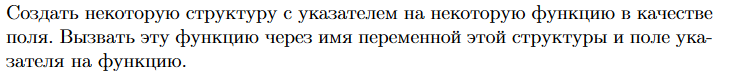
**Лабораторная работа 11**

**Задача 1**

1) Постановка задачи:



2) Математическая модель:

-

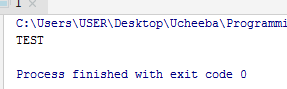
3) Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| **Имя** | **Смысл** | **Тип** |
| Ctruct | Структура | struct |
| s | Структура | struct |

4) Код программы:

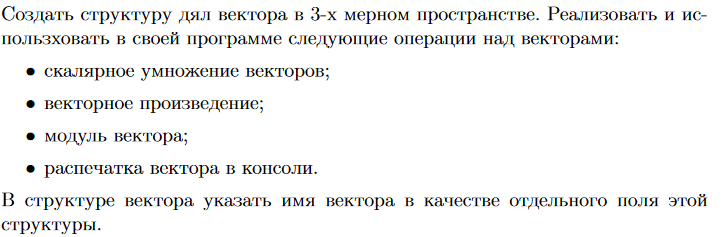
#include **<stdio.h>  
  
typedef struct** ctruct  
{  
 **int** (\*functia)();  
}strct;  
  
**void** vivod()  
{  
 printf(**"TEST\n"**);  
}  
  
**int** main(){  
 strct s;  
 s.functia = &vivod;  
 s.functia();  
 **return** 0;  
}

5) Результат работы:

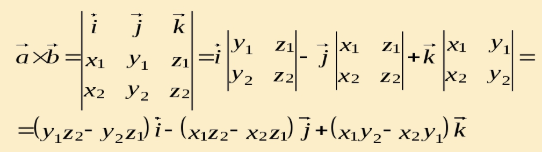


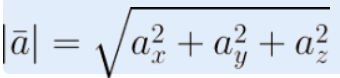
**Задача 2**

1) Постановка задачи:



2) Математическая модель:



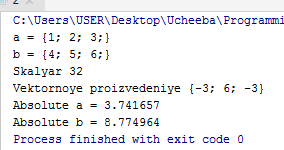
3) Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| **Имя** | **Смысл** | **Тип** |
| Vector | Структура вектора | struct |
| X,y,z | Векторное произведение | int |
| A,b | Векторы | struct |

4) Код программы:

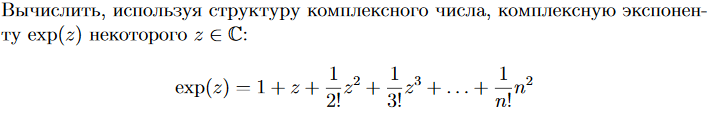
#include **<stdio.h>**#include **<stdlib.h>**#include **<math.h>  
  
struct** vector  
{  
 **char** name;  
 **int** x;  
 **int** y;  
 **int** z;  
};  
  
**int** main()  
{  
 **int** x,y,z;  
 **struct** vector a;  
 **struct** vector b;  
 a.name=**'a'**;  
 a.x=1;  
 a.y=2;  
 a.z=3;  
  
 b.name=**'b'**;  
 b.x=4;  
 b.y=5;  
 b.z=6;  
 printf\_s(**"%c = {%d; %d; %d;}\n"**, a.name, a.x, a.y, a.z);  
 printf\_s(**"%c = {%d; %d; %d;}\n"**, b.name, b.x, b.y, b.z);  
 printf\_s(**"Skalyar %d\n"**, a.x\*b.x + a.y\*b.y + a.z\*b.z);  
  
 x = a.y\*b.z - a.z\*b.y;  
 y = -(a.x\*b.z - a.z\*b.x);  
 z = a.x\*b.y - a.y\*b.x;  
  
 printf\_s(**"Vektornoye proizvedeniye {%d; %d; %d}\n"**, x, y, z);  
 printf\_s(**"Absolute %c = %lf\n"**, a.name, sqrt(a.x\*a.x + a.y\*a.y + a.z\*a.z));  
 printf\_s(**"Absolute %c = %lf"**, b.name, sqrt(b.x\*b.x + b.y\*b.y + b.z\*b.z));  
  
 **return** 0;  
}

5) Результат работы:



**Задача 3**

1) Постановка задачи:



2) Математическая модель:



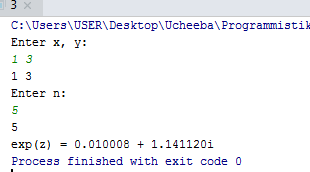
3) Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| **Имя** | **Смысл** | **Тип** |
| c | Структура комплексного числа | struct |
| n | Количество повторений | int |
| fac | Факториал | int |
| ugol | Угол | float |
| z | Структура | struct |
| I | Счетчик цикла | int |

4) Код программы:

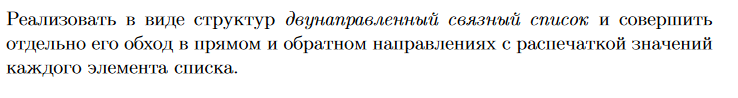
#include **<stdio.h>**#include **<stdlib.h>**#include **<math.h>  
  
struct** c  
{  
 **float** x;  
 **float** y;  
};  
  
**int** main() {  
 **int** n, fac;  
 **float** ugol;  
 **struct** c z;  
  
 printf\_s(**"Enter x, y: \n"**);  
 scanf\_s(**"%fac"**, &z.x);  
 scanf\_s(**"%fac"**, &z.y);  
  
 printf\_s(**"Enter n: \n"**);  
 scanf\_s(**"%d"**, &n);  
  
 ugol = z.y / z.x;  
 z.x = 1;  
 z.y = 1;  
 **for** (**int** i = 1; i < n; i++) {  
 fac = 1;  
 **if** (i != 0)  
 fac = fac \* i;  
 z.x += 1 / fac \* cos(i \* ugol);  
 z.y += 1 / fac \* sin(i \* ugol);  
 }  
 printf\_s(**"exp(z) = %f + %fi"**, z.x, z.y);  
 **return** 0;  
}

5) Результат работы:



**Задача 4**

1) Постановка задачи:



2) Математическая модель:

-

3) Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| **Имя** | **Смысл** | **Тип** |
| a | Число, записываемое в структуру | int |
| i | Счетчик цикла | int |
| n | Количество повторений и размер структуры | int |
| list | Список | struct |

4) Код программы:

#include **<stdio.h>**#include **<stdlib.h>  
  
typedef struct** spisok {  
 **int** value;  
 **struct** spisok \*next;  
 **struct** spisok \*prev;  
}spisok;  
  
spisok\* create(**int** n) {  
 spisok\* begin = malloc(**sizeof**(spisok));  
 **int** a;  
 printf\_s(**"Enter num: \n"**);  
 scanf\_s(**"%d"**,&a);  
 begin->value = a;  
 begin->prev = **NULL**;  
 spisok \*onStruct, \*onStructNext;  
 onStruct = begin;  
 **for** (**int** i = 1; i < n; i++) {  
 onStructNext = malloc(**sizeof**(spisok));  
 printf\_s(**"Enter next num: \n"**);  
 scanf\_s(**"%d"**,&a);  
 onStructNext->value = a;  
 onStruct->next = onStructNext;  
 onStructNext->prev = onStruct;  
 onStruct = onStructNext;  
 }  
 onStruct->next = **NULL**;  
 **return** begin;  
}  
  
**void** print(spisok \*begin) {  
 spisok \*onStruct = begin;  
 spisok \*onStructPosl;  
 **while** (onStruct != **NULL**) {  
 printf(**"%d "**, onStruct->value);  
 onStructPosl = onStruct;  
 onStruct = onStruct->next;  
 }  
 printf(**"\n"**);  
 **while** (onStructPosl != **NULL**) {  
 printf(**"%d "**, onStructPosl->value);  
 onStructPosl = onStructPosl->prev;  
 }  
}  
  
**void** freespisok(spisok \*begin) {  
 spisok \*onStructNext = begin->next;  
 **if** (onStructNext != **NULL**) {  
 freespisok(onStructNext);  
 }  
 free(begin);  
}  
  
**int** main() {  
 **int** n;  
 printf\_s(**"Enter lenght: \n"**);  
 scanf\_s(**"%d"**, &n);  
 spisok \*list = create(n);  
 print(list);  
 freespisok(list);  
 **return** 0;  
}

5) Результат работы:

