

Министерство науки и высшего образования РФ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский Авиационный Институт»
Национальный Исследовательский Университет

Факультет №8 «Информационные технологии и прикладная математика»
Кафедра 806 «Вычислительная математика и программирование»

КУРСОВОЙ ПРОЕКТ

По дисциплине «Практикум на ЭВМ»

Задание: «Разреженные матрицы»

Студент: Хренникова А. С.

Группа: М80-108-19

Преподаватель: Поповкин А. В.

Подпись:

Оценка:

Дата:

Содержание

Задание	3
Общий метод решения.....	5
Общие сведения о программе	6
Описание логической структуры программы	7
Описание переменных, функций, входные и выходные данные	8
Листинг программы	12
Пример работы программы.....	17
Заключение	23
Список использованных источников	24

Задание

Составить программу на языке Си с функциями/процедурами для обработки прямоугольных разреженных матриц с элементами вещественного типа, которая:

1. Вводит матрицы различного размера с одновременным размещением ненулевых элементов в разреженной матрице в соответствии с заданной схемой;
2. Печатает введенные матрицы во внутреннем представлении и в обычном виде;
3. Выполняет необходимые преобразования разреженных матриц (или вычисления над ними) путем обращения к соответствующим функциям;
4. Печатает результат преобразования во внутреннем представлении и в обычном виде.

В процедурах и функциях предусмотреть проверки и печать сообщений в случаях ошибок в задании параметров. Для отладки использовать матрицы, содержащие 5-10% ненулевых элементов, с максимальным числом элементов 100. Вариант схемы размещения матрицы определяется по формуле $((N + 3) \% 4) + 1$, где N -номер студента по списку в группе. Вариант преобразования определяется по формуле $((N-1) \% 11) + 1$. Вариант физического представления (1 - отображение на массив, 2 -отображение на динамические структуры) определяются по формуле $N \% 2 + 1$, где N номер студента по списку группы. В случае использования динамических структур индексы заменяются соответствующими ссылками.

Номер по списку N=23:

Вариант схемы размещения матрицы:

3. Три вектора:

CIP:	Индекс начала 1-ой строки в массивах PI и YE		Индекс начала 2-ой Строки	...	Индекс начала N-ой Строки
PI:	Номер столбца	Номер столбца	Номер столбца	...	Номер Столбца
	↓	↓	↓		↓
YE:	Значение	Значение	Значение	...	Значение

Вариант преобразования №1:

- Определить максимальный по модулю элемент матрицы и разделить на него все элементы строки, в которой он находится. Если таких элементов несколько, обработать каждую строку, содержащую такой элемент.

Вариант физического представления №2:

- Отображение на динамические структуры.

Общий метод решения

Работа осуществляется с разреженными матрицами, представленными в отдельном текстовом файле. В зависимости от выбора действия пользователем программа выполняет те или иные функции (считывание матрицы из файла, вывод исходной матрицы в естественном виде и во внутреннем представлении, выполнение необходимых преобразований матриц, вывод результата преобразований и т.д.). При считывании элементов матрицы из файла программа одновременно размещает ненулевые элементы в соответствии с заданной схемой размещения.

Решение строится на реализации структуры, состоящей из двух односвязных списков для массива значений и массива номеров столбцов и динамического массива указателей. В первом односвязном списке хранятся значения ненулевых элементов матрицы, во втором хранятся номера столбцов соответствующих элементов из первого списка. В динамическом массиве указателей хранятся ссылки на элементы начала i -той строки в предыдущих двух списках. Если i -тая строка состоит полностью из нулевых элементов, то в массив записывается значение NULL, при выводе индексов NULL заменяется на «-1».

Общие сведения о программе

Необходимое программное и аппаратное обеспечение: ОС семейства UNIX (Linux Ubuntu), среда программирования Си (язык Си, компилятор gcc), процессор с 64-битной архитектурой (как на лабораторном компьютере).

Система программирования: GUN C.

Местонахождение файлов на домашнем компьютере: /home/lina_tucha/dir/kp7. Сам файл компилируется с помощью написания «gcc kurs7.c && ./a.out» в командной строке интерпретатора команд.

Описание логической структуры программы

Программа представляет собой файл исходного кода языка Си. Вначале программа просит ввести имя файла, из которого в дальнейшем будет производиться считывание матриц. Потом идет вызов функции *menu()*, которая выводит информацию о возможностях программы, а именно какие действия она может совершать. Пользователь выбирает, какое действие он хочет выполнить и вводит цифру, которой соответствует та или иная операция. В зависимости от введенного числа программа начинает выполнять определенное действие.

- Если введена цифра «1», то вызывается функция *enter()*, которая производит считывание текущей матрицы из файла. Одновременно со считыванием происходит размещение ненулевых элементов матрицы в трёх векторах, а именно в структуре *matrix*.
- Если введена цифра «2», то вызывается функция *output()*, которая производит вывод последней считанной матрицы в обычном (естественном) виде и во внутреннем представлении.
- Если введена цифра «3», то вызывается функция *transform()*, которая производит заданные преобразования с последней считанной матрицей, а именно ищет в ней максимальный по модулю элемент и делит на него все элементы строки, в которой он находится. Если таких элементов в матрице несколько, то преобразовывается каждая строка матрицы, содержащая такой элемент.
- Если введена цифра «4», то программа завершает свою работу и закрывается.

Описание переменных, функций, входные и выходные данные

Описание используемых структур

1) Название структуры: node

Описание:

```
typedef struct node {  
    struct node *next;  
    double value;  
}node;
```

Данная структура является связным списком и используется для представления вектора значений и вектора номеров столбцов.

2) Название структуры: matrix

Описание:

```
typedef struct matrix{  
    node **cip;  
    node *pi;  
    node *ye;  
}matrix;
```

Данная структура реализует схему размещения матрицы по трем векторам.

Описание переменных программы

Таблица 1 – Описание переменных программы

Тип переменной	Название переменной	Значение переменной
matrix	m	переменная структуры <i>matrix</i>
int	stlb	количество столбцов в считанной матрице
int	str	количество строк в считанной матрице
int	b	динамический массив в котором хранятся индексы
int	bb	динамический массив в котором хранятся индексы
int	n	хранит цифру, которую вбил пользователь
FILE	file	переменная хранит указатель на объект файла
int	count2	кол-во элементов массива ссылок

Описание функций программы

1) *enter()*

Данная функция производит считывание элементов текущей матрицы из файла и размещает её ненулевые элементы в структуре *matrix*, согласно схеме размещения.

Таблица 2 - Описание переменных данной функции

Тип переменной	Название переменной	Значение переменной
double	elem	считываемый элемент
int	column	номер столбца считавшегося элемента
int	i	количество элементов в массиве ссылок
char	s	считываемый символ
int	count	количество считанных элементов
int	check	хранит информацию об открытии файла
char	S2	последний считанный символ

2) *transform()*

Производит преобразования с последней считанной матрицей из файла. Определяет максимальный по модулю элемент матрицы и делит на него все элементы строки, в которой он находится. Если таких элементов несколько, обрабатывает каждую строку, содержащую такой элемент.

Таблица 3 - Описание переменных данной функции

Тип переменной	Название переменной	Значение переменной
int	i	кол-во строк в текущей матрицей
int	j	номер столбца считавшегося элемента
int	str	количество элементов в массиве ссылок
int	stlb	считываемый символ
double	max1[]	максимальные по модулю элементы
double	max2[]	номера строк в которых находятся максимальные эл-ты
int	ind	кол-во найденных максимальных по модулю элементов

Таблица 3. Продолжение

double	num	элемент из вектора значений
node	ye	указатель на начало списка вектора значений
node	pi	указатель на начало списка вектора номеров столбцов
double	p	элемент из вектора номеров столбцов

3) *output()*

Выводит либо последнюю считанную матрицу из файла, либо новую матрицу, полученную в результате преобразования в обычном виде и во внутреннем представлении.

Таблица 4 - Описание переменных данной функции

Тип переменной	Название переменной	Значение переменной
int	n	кол-во уже выведенных элементов
int	c	количество элементов в массиве индексов
node	p1	указатель на начало списка вектора значений
node	p2	указатель на начало списка вектора номеров столбцов

4) *print(node *head)*

Получает параметром указатель на начало списка. Выводит все элементы списка.

5) *menu()*

Выводит информацию о возможностях программы (какие действия с матрицами она может выполнять) с краткой инструкцией по использованию.

6) *init()*

Получает в качестве параметра значение переменной, которую нужно добавить в список. Создаёт указатель *lst* типа *node* на начало списка,

записывает в него значение и возвращает данный указатель. Возвращаемый тип данной функции – `struct node*`.

7) *push()*

Получает в качестве параметров, указатель на последний элемент и значение элемента, который нужно добавить в список. Создает узел списка *tmp* типа *node* и записывает в него значение элемента и возвращает указатель на него. Возвращаемый тип функции - `struct node*`.

8) Функции сообщений об ошибке :

- *error1()* - выводит сообщение об ошибке , если во время преобразования матрицы было выявлено, что максимальный по модулю элемент 0, а на 0 делить нельзя.
- *error2()*- выводит сообщение об ошибке, если файл, из которого должны считываться матрицы, не может быть открыт.
- *error3()*- выводит сообщение об ошибке, когда достигнут конец файла, а пользователь хочет считать матрицу.

Входные данные

Числа от 1 до 4, соответствующие той операции, которую хочет выполнить пользователь.

«1» - считать матрицу

«2» - вывести матрицу

«3» - выполнить преобразования с матрицей

«4» - завершить выполнение программы

Выходные данные

Вывод меню, а также в зависимости от введенной цифры либо сообщение об успешном выполнении операции, либо матрицу, напечатанную в обычном (естественном) виде и во внутреннем представлении.

Листинг программы

```
#include <stdio.h>
#include <malloc.h>
#include <stdlib.h>
#include <math.h>

void error1(){printf("The matrix cannot be transformed \n\nThe maximum modulo element is
zero\n");};
void error2(){ printf("Can't open file\n"); };
void error3(){printf("No matrix anymore\n\n");};

typedef struct node {

struct node *next;
double value;
}node;

struct node* init(double a){
    node *lst=(node*)malloc(sizeof(node));
    lst->value=a;
    lst->next=NULL;
    return(lst);

};

struct node * push(node *lst, double num)
{
    node *tmp= (node*)malloc(sizeof(node));
    lst->next = tmp;
    tmp->value = num;
    tmp->next = NULL;
    return(tmp);
}

void print(node *head) {
if(head==NULL)printf("empty");
else{
    while (head) {
        if(head->value < 0.0){printf("%-6.2lf ", head->value);}
        else printf("%-5.2lf ", head->value);
        head = head->next;}
    }printf("\n");
}

void printt(node *head, node *h) {
if(head==NULL)printf("empty");
else{
    while (head) {
        if(h->value < 0.0){printf("%-6.0lf ", head->value);}
        else printf("%-5.0lf ", head->value);
        head = head->next; h=h->next;
    }
}
```

```

    }
    printf("\n");
}

typedef struct matrix{
    node **cip;
    node *pi;
    node *ye;
}matrix;

matrix m;
int stlb=0,str=0,*b=NULL,res;
int *bb=NULL,count2;
FILE* file;

int enter()
{
    double elem;
    m.ye=NULL; m.cip=NULL; m.pi=NULL;
    int column=1,i=0,check=1,count=0;
    stlb=0,str=0;
    char s,s2;
    m.cip=(node**)malloc(i*sizeof(node*));
    node *p1,*p2 ;
    if(feof(file)!=0){error3();return 1;}
    while (fscanf(file,"%lf",&elem)!=EOF)
    {
        if(elem!=0){
            if(count==0)
            {
                m.ye=init(elem);
                m.pi=init(column);
                p1=m.ye;
                p2=m.pi;
            }
            if(check)
            {
                m.cip=(node**)realloc(m.cip,(i + 1) * sizeof(node*));
                b=(int*)realloc(b, (i + 1) * sizeof(int));
                *(m.cip+i)=p1;
                *(b+i)=count;
                i++;
            }
            if(count!=0){
                p1=push(p1,elem);
                p2=push(p2,column);}
            count++;
            check=0;}
        column++;
        fscanf(file,"%c",&s);
        if(s=='\n')
        {stlb=column-1;column=1;check=1;str++;

```

```

        if(i<str){
            m.cip=(node**)realloc(m.cip,(i + 1) * sizeof(node*));
            b=(int*)realloc(b, (i + 1) * sizeof(int));
            *(m.cip+i)=NULL;
            *(b+i)=-1;
            i++;}
        fscanf(file,"%c",&s2);
        if(s2=='\n')break; } }
return 0;}

```

```

int output( )
{ int i=0,n=1,c=-1;
  int *bbb;
  node *p1, *p2;
  p1=m.ye;p2=m.pi;bbb=b;
  printf("Matrix in the usual form :");
  for(int k=0; k<str; k++){
      n=1;
      printf("\n");c++;
      while(n<=stlb){
          if(p2 && n==p2->value && *(bbb+c)!=-1)
          {printf("%6.2lf ", p1->value);
            p1=p1->next;
            p2=p2->next;}
          else
          {printf("%6.2lf ", 0.0);}
          n++;}
      }
  for(int k=n; k<stlb+1; k++)printf("%6.2lf ", 0.0);
  printf("\n");
  printf("\n");
  printf("Matrix in internal view : \n");
  if(m.ye==NULL)printf("All matrix elements are zero\n");
  printf("CIP : ");
  int k=0;
  while(printf("%d ",*(b+k))>0 && k<str-1)k++;
  printf("\n");
  printf( "PI : ");printt(m.pi,m.ye);
  printf( "YE : ");print(m.ye);
return 0; }

```

```

int transform( ){
    double max1[30]; int max2[30];
    int ind=0;
    for(int i=0; i<30; i++){ max1[i]=0.0; max2[i]=0;}
    int i=str,j=stlb,ind2=0,ch=0;
    double n,p,num;
    node *pi=m.pi,*ye=m.ye,*pi1=m.pi,*ye1=m.ye;

    int k;
    for( k=1; k<=i; k++) // пока не закончились строки
    {

```

```

n=1;
while(pi1 && n<=j)
{
    p=pi1->value; //столбец
    num=ye1->value; // значение
    if(p==n && *(b+k-1)!=-1) // если это всё еще та же строка
    {
        if( abs(num)>abs(max1[ind]) )
        {
            for( int j=0; j<=ind; j++) { max1[j]=0.0 ; max2[j]=0;}
            ind=0;
            max1[0]=num; max2[0]=k;
        }
        else if( abs(num)==abs(max1[ind]) && k!=max2[ind])
        {
            ind++;
            max1[ind]=num; max2[ind]=k;
        }
    }

    pi1=pi1->next;
    ye1=ye1->next;}
n++;}
}
if(max1[0]==0.0){ error1(); return 1;};
for( k=1; k<=i; k++) // пока не закончились строки
{
    if(ch){ind2++;}
    ch=0;
    n=1;
    while(pi && n<=j)
    {
        p=pi->value; //столбец
        num=ye->value; // значение
        if(p==n && *(b+k-1)!=-1) // если это всё еще та же строка
        {
            if(k==max2[ind2]) // в строке есть элемент
            {
                ye->value=num/max1[ind2];
                ch=1;
            }
        }
        pi=pi->next;
        ye=ye->next;}
    n++;}
}

return 0;}
int menu()
{
    printf("Print 1 if you want to enter matrix \n");
    printf("Print 2 if you want to print matrix \n");
    printf("Print 3 if you want to transform the matrix\n");
    printf("Print 4 if you want to exit the program\n");

```

```

    }

int main()
{
    int n,chk=1;
    printf("Print name of file: ");
    char *a[30];
    scanf("%s",a);
    file = fopen(a,"r");
    if (!file) {
        error2();
        return 1; }
    menu();
    scanf("%d",&n);
    while(1){
        if(n==1)
        { if(!enter())printf("The matrix was successfully read\n\n");
          chk=0;
        }
        if(n==2){ output(); printf("\n");}
        if(n==3)
        {
            if(!transform())printf("The matrix was successfully transformed\n\n");
        }
        if (n==4)
        { printf("The program is closed\n"); return 0;}
        printf("1-enter, 2-print, 3-transform, 4-exit\n");
        scanf("%d",&n);
    }
}

```


Пример работы программы

```
lina_tucha@LAPTOP-44CRFC1U:~/dir/kp7$ gcc kurs7.c && ./a.out
```

Print name of file: matr

Print 1 if you want to enter matrix

Print 2 if you want to print matrix

Print 3 if you want to transform the matrix

Print 4 if you want to exit the program

1

The matrix was successfully read

1-enter, 2-print, 3-transform, 4-exit

2

Matrix in the usual form :

0.00 0.00 0.00 1.00

2.00 0.00 0.00 0.00

2.00 0.10 1.20 0.20

0.10 0.00 0.30 0.40

0.00 0.00 0.00 2.00

Matrix in internal view :

CIP : 0 1 2 6 9

PI : 4 1 1 2 3 4 1 3 4 4

YE : 1.00 2.00 2.00 0.10 1.20 0.20 0.10 0.30 0.40 2.00

1-enter, 2-print, 3-transform, 4-exit

3

The matrix was successfully transformed

1-enter, 2-print, 3-transform, 4-exit

2

Matrix in the usual form :

0.00 0.00 0.00 1.00

1.00 0.00 0.00 0.00

1.00 0.05 0.60 0.10

0.10 0.00 0.30 0.40

0.00 0.00 0.00 1.00

Matrix in internal view :

CIP : 0 1 2 6 9

PI : 4 1 1 2 3 4 1 3 4 4

YE : 1.00 1.00 1.00 0.05 0.60 0.10 0.10 0.30 0.40 1.00

1-enter, 2-print, 3-transform, 4-exit

1

The matrix was successfully read

1-enter, 2-print, 3-transform, 4-exit

2

Matrix in the usual form :

1.00 0.00 0.00 4.00 5.00

0.00 7.00 -12.00 0.00 10.00

11.00 12.00 0.00 0.00 0.00

Matrix in internal view :

CIP : 0 3 6

PI : 1 4 5 2 3 5 1 2

YE : 1.00 4.00 5.00 7.00 -12.00 10.00 11.00 12.00

1-enter, 2-print, 3-transform, 4-exit

3

The matrix was successfully transformed

1-enter, 2-print, 3-transform, 4-exit

2

Matrix in the usual form :

1.00 0.00 0.00 4.00 5.00

0.00 -0.58 1.00 0.00 -0.83

0.92 1.00 0.00 0.00 0.00

Matrix in internal view :

CIP : 0 3 6

PI : 1 4 5 2 3 5 1 2

YE : 1.00 4.00 5.00 -0.58 1.00 -0.83 0.92 1.00

1-enter, 2-print, 3-transform, 4-exit

1

The matrix was successfully read

1-enter, 2-print, 3-transform, 4-exit

2

Matrix in the usual form :

0.00 0.00

0.00 0.00

0.00 0.00

Matrix in internal view :

All matrix elements are zero

CIP : -1 -1 -1

PI : empty

YE : empty

1-enter, 2-print, 3-transform, 4-exit

3

The matrix cannot be transformed

The maximum modulo element is zero

1-enter, 2-print, 3-transform, 4-exit

1

The matrix was successfully read

1-enter, 2-print, 3-transform, 4-exit

2

Matrix in the usual form :

0.00 30.00
6.00 0.00
0.00 0.00

Matrix in internal view :

CIP : 0 1 -1
PI : 2 1
YE : 30.00 6.00

1-enter, 2-print, 3-transform, 4-exit
3

The matrix was successfully transformed

1-enter, 2-print, 3-transform, 4-exit
2

Matrix in the usual form :

0.00 1.00
6.00 0.00
0.00 0.00

Matrix in internal view :

CIP : 0 1 -1
PI : 2 1
YE : 1.00 6.00

1-enter, 2-print, 3-transform, 4-exit
1

The matrix was successfully read

1-enter, 2-print, 3-transform, 4-exit
2

Matrix in the usual form :

0.00 0.00 -7.00
7.00 5.00 7.00

Matrix in internal view :

CIP : 0 1
PI : 3 1 2 3
YE : -7.00 7.00 5.00 7.00

1-enter, 2-print, 3-transform, 4-exit
3

The matrix was successfully transformed

1-enter, 2-print, 3-transform, 4-exit
2

Matrix in the usual form :

0.00 0.00 1.00
1.00 0.71 1.00

Matrix in internal view :

CIP : 0 1
PI : 3 1 2 3
YE : 1.00 1.00 0.71 1.00

1-enter, 2-print, 3-transform, 4-exit
1
The matrix was successfully read

1-enter, 2-print, 3-transform, 4-exit
2
Matrix in the usual form :
0.00
0.00
1.50
6.40

Matrix in internal view :
CIP : -1 -1 0 1
PI : 1 1
YE : 1.50 6.40

1-enter, 2-print, 3-transform, 4-exit
3
The matrix was successfully transformed

1-enter, 2-print, 3-transform, 4-exit
2
Matrix in the usual form :
0.00
0.00
1.50
1.00

Matrix in internal view :
CIP : -1 -1 0 1
PI : 1 1
YE : 1.50 1.00

1-enter, 2-print, 3-transform, 4-exit
1
The matrix was successfully read

1-enter, 2-print, 3-transform, 4-exit
2
Matrix in the usual form :
9.00 4.50 2.10
5.00 0.00 0.00
0.00 0.00 0.00
1.50 3.50 7.80
2.10 0.00 1.00

Matrix in internal view :

CIP : 0 3 -1 4 7
PI : 1 2 3 1 1 2 3 1 3
YE : 9.00 4.50 2.10 5.00 1.50 3.50 7.80 2.10 1.00

1-enter, 2-print, 3-transform, 4-exit

3

The matrix was successfully transformed

1-enter, 2-print, 3-transform, 4-exit

2

Matrix in the usual form :

1.00 0.50 0.23
5.00 0.00 0.00
0.00 0.00 0.00
1.50 3.50 7.80
2.10 0.00 1.00

Matrix in internal view :

CIP : 0 3 -1 4 7
PI : 1 2 3 1 1 2 3 1 3
YE : 1.00 0.50 0.23 5.00 1.50 3.50 7.80 2.10 1.00

1-enter, 2-print, 3-transform, 4-exit

1

The matrix was successfully read

1-enter, 2-print, 3-transform, 4-exit

2

Matrix in the usual form :

0.00 0.00 0.00 1.00 0.00
2.00 0.10 1.20 0.20 0.30
2.00 2.00 0.00 0.00 2.00
0.00 0.00 0.00 0.00 0.00
0.00 0.00 1.00 2.00 0.00

Matrix in internal view :

CIP : 0 1 6 -1 9
PI : 4 1 2 3 4 5 1 2 5 3 4
YE : 1.00 2.00 0.10 1.20 0.20 0.30 2.00 2.00 2.00 1.00 2.00

1-enter, 2-print, 3-transform, 4-exit

3

The matrix was successfully transformed

1-enter, 2-print, 3-transform, 4-exit

2

Matrix in the usual form :

0.00 0.00 0.00 1.00 0.00
1.00 0.05 0.60 0.10 0.15
1.00 1.00 0.00 0.00 1.00
0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.50 1.00 0.00

Matrix in internal view :

CIP : 0 1 6 -1 9

PI : 4 1 2 3 4 5 1 2 5 3 4

YE : 1.00 1.00 0.05 0.60 0.10 0.15 1.00 1.00 1.00 0.50 1.00

1-enter, 2-print, 3-transform, 4-exit

1

No matrix anymore

1-enter, 2-print, 3-transform, 4-exit

4

The program is closed

lina_tucha@LAPTOP-44CRFC1U:~/dir/kp7\$ cat matr

0.0 0.0 0.0 1.0

2.0 0.0 0.0 0.0

2.0 0.1 1.2 0.2

0.1 0.0 0.3 0.4

0.0 0.0 0.0 2.0

1.0 0.0 0.0 4.0 5.0

0.0 7.0 -12.0 0.0 10.0

11.0 12.0 0.0 0.0 0.0

0.0 0.0

0.0 0.0

0.0 0.0

0.0 30.0

6.0 0.0

0.0 0.0

0.0 0.0 -7.0

7.0 5.0 7.0

0.0

0.0

1.5

6.4

9.0 4.5 2.1

5.0 0.0 0.0

0.0 0.0 0.0

1.5 3.5 7.8

2.1 0.0 1.0

0.00 0.00 0.00 1.00 0.00

2.00 0.10 1.20 0.20 0.30

2.00 2.00 0.00 0.00 2.00

0.00 0.00 0.00 0.00 0.00

0.00 0.00 1.00 2.00 0.00

Заключение

Благодаря данному заданию курсового проекта я научилась обрабатывать разреженные прямоугольные матрицы с элементами вещественного типа и размещать ненулевые элементы этих матриц в соответствии с различными схемами внутреннего представления разреженных матриц. Так же я научилась такому варианту физического представления, как отображение на динамические структуры. Научилась реализовывать односвязные списки.

Список использованных источников

1. РосДиплом, Оформление таблиц в дипломной работе, особенности и требования ГОСТ/Электронный диплом/Режим доступа: <https://www.rosdiplom.ru/rd/pubdiplom/view.aspx?id=288>
2. Диплом Журнал, Оформление курсовой работы по ГОСТу 2019(образец)/Электронный диплом/Режим доступа: <https://journal.duplom.ru/kursovaya/oformlenie-kursovoj-raboty-po-gostu-2019-obrazec/>
3. Vyuchit.work – универсальная методичка/Электронный диплом/Режим доступа: <https://vyuchit.work/samorazvitie/sekretyi/oformlenie-risunkov-po-gostu.html>
4. Керниган, Брайан У., Ритчи, Деннис М. Язык программирования С, 2-е издание. :Пер. с англ. – М. : Издательский дом «Вильямс», 2009. – 304 с. : ил. – Парал. тит. англ.
5. Берзтисс А. Т., Структуры данных: Пер. с англ. В. Л. Грейсуха и И. С. Левиной. - М.: Статистика, 1974. - 408 с..
6. Р. Тьюарсон, Разреженные матрицы: Пер. с англ. Э.М. Пейсаховича ; Под ред. Х.Д. Икрамова. - Москва : Мир, 1977. - 189 с. : ил..
7. Писсанецки С.,Технология разреженных матриц: Пер. с англ.—М.: Мир, 1988. —410 с , ил..