



برنامه نویسی پیشرفته – ترم 4001

تمرین اول : کار با ماتریس ها

استاد : مهندس بطحائیان

محمدحسین الطافی - 9912358004

شرح تمرین

تمرین اول درباره ی کار با ماتریس ها , دریافت آن ها , تشخیص نوع آنها (مثل قطری بودن , بالامثلثی بودن و...) , ذخیره کردن و حذف ماتریس است.

ساختار برنامه

- در این برنامه , یک استراکت به نام Matrix تعریف شده که شامل:

1. String name

2. String **mat

3. Int satr

4. Int sotun

است. اشاره گر دو بعدی از نوع استرینگ به این خاطر است که استرینگ میتواند شامل همه ی تایپ ها باشد و هر جا نیاز شد می توان آن را به نوع Int یا double تبدیل کرد.

- وکتوری از این استراکت ها به نام matrixes تعریف شده و وقتی که ماتریس جدیدی از کاربر دریافت شد , به این وکتور push_back می شود.

- برای تشخیص نوع دستور ورودی , تابعی به نام string_split تعریف شده که دستور را با توجه به فاصله هایی که در آن وجود دارد, تکه تکه کرده و در یک وکتور به نام optiontokens ذخیره می کند.

- از این تکه دستور ها برای فراخوانی توابع مربوط به هر دستور استفاده می شود.

توابع برنامه

```
vector<string> string_split(const string &option, char delimiter)
```

این تابع استرینگ دریافتی را با توجه به `delimiter` که دریافت می کند، تکه تکه می کند و داخل یک وکتور قرار داده و این وکتور خروجی تابع است.

```
void addmatrix(string &name, const int satr, const int sotun, string initializes = "")
```

این تابع برای اضافه کردن ماتریس با مقادیری که دریافت می کند استفاده می شود.

```
bool namecheckerForAddMatrix(string name)
```

این تابع در تابع قبل برای چک کردن تکراری نبودن اسم ورودی استفاده می شود.

```
bool is_number(const string &str)
```

این تابع برای چک کردن عدد بودن خانه ای از ماتریس برای دستوراتی که فقط روی اعداد اجرا می شوند، استفاده می شود.

```
bool is_diagonal(const int &len, string **mat)
```

از این تابع برای چک کردن قطری بودن ماتریس استفاده می شود.

```
bool is_upper_triangular(const int &len, string **mat)
```

از این تابع برای چک کردن بالا مثلثی بودن ماتریس استفاده می شود.

```
bool is_lower_triangular(const int &len, string **mat)
```

از این تابع برای چک کردن پایین مثلثی بودن ماتریس استفاده می شود.

```
bool is_triangular(const int &len, string **mat)
```

این تابع با استفاده از دو تابع قبل، نوع مثلثی بودن ماتریس را مشخص می کند.

```
bool is_identity(const int &len, string **mat)
```

از این تابع برای چک کردن همانی بودن ماتریس استفاده می شود.

```
bool is_normal_symmetric(const int &len, string **mat)
```

از این تابع برای چک کردن متقارن بودن ماتریس استفاده می شود.

```
bool is_skew_symmetric(const int &len, string **mat)
```

از این تابع برای چک کردن پادمتقارن بودن ماتریس استفاده می شود.

```
bool is_symmetric(const int &len, string **mat)
```

این تابع با استفاده از دو تابع قبل، نوع تقارن ماتریس را مشخص می کند.

```
void inverse(const int &len, string **mat, string newmatrixname = "")
```

از این تابع برای معکوس کردن ماتریس (در این تمرین منظور از معکوس، ترانزپاز ماتریس مربعی است) استفاده می شود.

```
void showmatrix(const int &satr, const int &sotun, string **const &mat, string &name)
```

از این تابع برای نمایش ماتریس استفاده می شود.

```
void deletematrix(string **mat, int satr, int sotun, string Name)
```

از این تابع برای حذف ماتریس استفاده می شود.

```
void change(string **mat, const int &satr, const int &sotun, const string &newvalue)
```

از این تابع برای تغییر محتوای یک خانه از یک ماتریس استفاده می شود.

```
void appinfo()
```

این تابع لیستی از دستورات برنامه که کاربر می تواند وارد کند را نمایش می دهد.

```
bool option_tokenizer(const string option)
```

این تابع نوع دستور ورودی را تشخیص می دهد و تابع مربوط به آن را فراخوانی می کند.

```
int main()
```

در تابع main , دستور دریافت می شود و به تابع option_tokenizer ارسال می شود.

محمدحسین الطافی

9912358004