

③ファイル入力によりデータを構造体に入れる ～構造体の定義、宣言～

```
#include <stdio.h>
```

```
struct planets{  
  
    char name[64];  
    long long diameter;  
    long long distance; } 値が大きい可能性あり  
};
```

構造体の定義

```
struct planets a[1000];
```

構造体配列の宣言
planets型の構造体を1000個作る

③ファイル入力によりデータを構造体に入れる ～ファイルからの読みこみ～

```
int main(){
```

```
FILE* f;  
  
if ((f = fopen("planets.txt","r"))==NULL){  
    printf("ファイルを開けませんでした");  
    exit(1);  
}
```

ファイルを開く。

もし、ファイルを開けなければ、
強制終了。

EOFが出るまでファイルのデータを読み込んでいく。

構造体のメンバには” 構造体 . メンバ” でアクセス

```
int i=0;  
while( fscanf(f, "%s %lld %lld", &(a[i].name),&(a[i].diameter), &(a[i].distance))!= EOF ){  
    i++;  
}  
  
fclose(f);
```

⑥ 行列とベクトルの掛け算 $Ax = y$ (動的配列の利用例) ～ポインタを宣言～

Aの行数列数をコンパイル後に決めたい → 動的配列が必要

```
int row, col;
```

```
double ** A, *x, *y;
```

まず、ポインタを宣言。

2次元配列が取りたければダブルポインタ。

```
printf("行数を入力してください\n");  
scanf("%d", &row);
```

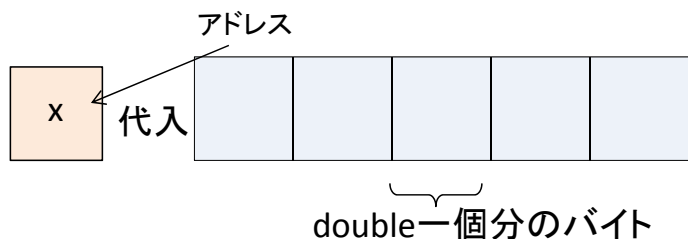
```
printf("列数を入力してください\n");  
scanf("%d", &col);
```

⑥ 行列とベクトルの掛け算 $Ax = y$ (動的配列の利用例) ～一次元動的配列～

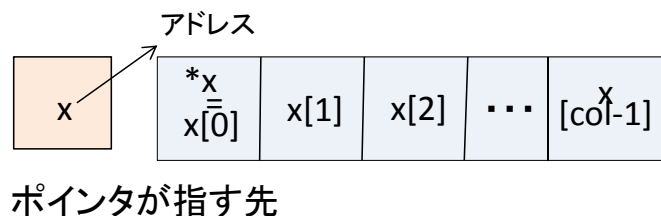
```
x = (double*)malloc( col * sizeof(double) );  
y = (double*)malloc( row * sizeof(double) );
```

(i)

確保したメモリ



(ii)



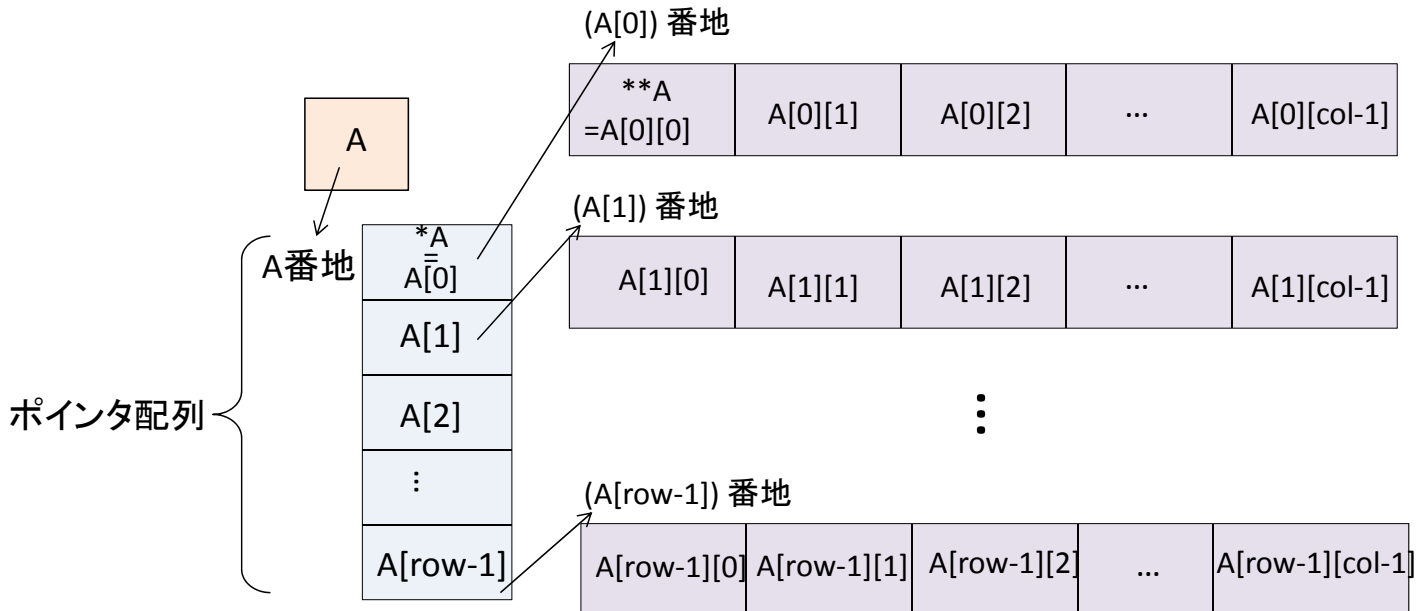
i) x用のメモリを、(行数) × (double一個に必要なバイト数) 確保し、そのメモリブロックの先頭へのポインタをxに代入。

ii) xが指すのはx[0]のアドレスで、確保したメモリが連続であることから、x[0], x[1], ... ,x[col-1] とアクセスすることができる。
(*x = x[0])

⑥ 行列とベクトルの掛け算 $Ax = y$ (動的配列の利用例) ～二次元動的配列～

```
A = (double**) malloc( row * sizeof(double*) );
for(int i=0; i<row; i++)
    A[i] = (double*) malloc( col * sizeof(double) );
```

← (行数) × (double へのポインタ
一個に必要なバイト数) のメモリ
を確保. その先頭アドレスをAに
代入



⑥ 行列とベクトルの掛け算 $Ax = y$ (動的配列の利用例) ～初期化と計算～

配列の初期化

```
for(int i=0; i<col; i++)
    x[i] = 0.0;
for(int i=0; i<row; i++)
    y[i] = 0.0;

for(int i=0; i<row; i++){
    for(int j=0; j<col; j++){
        A[i][j] = 0.0;
    }
}
```

行列の計算

```
printf("要素を入力してください\n");
for(int i=0; i<row; i++)
    for(int j=0; j<col; j++)
        scanf("%lf", &A[i][j]);

printf("ベクトルを入力してください\n");
for(int i=0; i<col; i++)
    scanf("%lf", &x[i]);

for(int i=0; i<row; i++)
    for(int j=0; j<col; j++)
        y[i] += A[i][j] * x[j];

for(int i=0; i<row; i++)
    printf("%lf\n", y[i]);
```

⑥ 行列とベクトルの掛け算 $Ax = y$ (動的配列の利用例) ～メモリ解放～

```
for(int i=0; i<row; i++)  
    printf("%lf¥n", y[i]);
```

結果の出力

```
free(x);  
free(y);  
  
for(int i=0; i<row; i++)  
    free(A[i]);  
  
free(A);
```

メモリの解放

free(ポインタ) で解放.
多次元の場合は, 後で確保
した領域から解放.

③のコード

```
int main(){  
  
    FILE* f;  
    if ((f = fopen("planets.txt", "r"))==NULL){  
        printf("ファイルを開けませんでした");  
        exit(1);  
    }  
  
    int i=0;  
  
    while( fscanf(f, "%s %lld %lld", &(a[i].name), &(a[i].diameter), &(a[i].distance)) != EOF ){  
        i++;  
    }  
  
    fclose(f);  
  
    int j;  
  
    for(j=0; j<i; j++)  
        printf("name: %s diameter: %lld distance: %lld¥n", a[j].name, a[j].diameter, a[j].distance);  
}
```

⑥のコード(1)

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

int main(){

int row, col;
double ** A, *x, *y;

printf("行数を入力してください\n");
scanf("%d", &row);

printf("列数を入力してください\n");
scanf("%d", &col);

x = (double*)malloc( col * sizeof(double) );
y = (double*)malloc( row * sizeof(double) );

A = (double**) malloc( row * sizeof(double*) );

for(int i=0; i<row; i++)
A[i] = (double*) malloc( col * sizeof(double) );
```

⑥のコード(2)

```
for(int i=0; i<col; i++)
x[i] = 0.0;

for(int i=0; i<row; i++)
y[i] = 0.0;

for(int i=0; i<row; i++)
for(int j=0; j<col; j++)
A[i][j] = 0.0;

printf("要素を入力してください\n");
for(int i=0; i<row; i++)
for(int j=0; j<col; j++)
scanf("%lf", &A[i][j]);

printf("ベクトルを入力してください\n");
for(int i=0; i<col; i++)
scanf("%lf", &x[i]);
```

⑥のコード(3)

```
for(int i=0; i<row; i++)
  for(int j=0; j<col; j++)
    y[i] += A[i][j] * x[j];

for(int i=0; i<row; i++)
  printf("%lf\n",y[i]);

free(x);
free(y);

for(int i=0; i<row; i++)
  free(A[i]);

free(A);
}
```