

sizeof という関数を使うと、様々な変数の大きさ（記憶容量）を知ることができます。

たとえば以下のように書くと、int（整数型）と整数型のポインタのサイズが印字されます。ポインタのサイズや整数等のサイズは計算機やコンパイラやOSなどの処理系によって変わることがあるので、自分のつかっている環境でのそれぞれの型のサイズを一度確認しておいて下さい。

```
===== size.c =====
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    printf("%lu\n",sizeof(int));          // 整数型のサイズ 型は 長い (8バイト) 符号なし整数 long unsigned int → lu
    printf("%lu\n",sizeof(int *));        // 整数型のポインタのサイズ

    double d;                             // 倍精度実数型変数を宣言
    double dm[10];                         // 倍精度実数型の配列を宣言
    printf("%lu\n",sizeof(d));             // dの変数のサイズをプリント
    printf("%lu\n",sizeof(dm));           // dm の変数のサイズをプリント
}
===== size.c =====
```

プログラムを拡張して、課題用紙（別紙）にある変数型や変数のサイズを出力し、プリント（別紙）に記入してください

## 基本課題2. (別紙に記入して提出)

printf文の型変換子に %p を用いるとポインタ変数の値（アドレス値）を出力することができます。以下のような短いプログラムを記述して、コンパイルして何回か実行し、実行するごとに、変数 i の置かれる場所（アドレス値 16進数で示されます）が変化することを確認してプリント（別紙）に記入してください

```
===== pointer.c =====
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    int i=0;
    printf("%p\n",&i);
}
===== pointer.c =====
```

## 基本課題3. プログラム名 revcomp (必須：提出期限、6月9日正午、ただしできるだけ講義時間内に提出)

課題に使うファイルは **git clone <https://github.com/NGS-maps/PracticeGenome>** としてダウンロードしておく。

塩基配列を記述したfastaファイル、(Bacillus.fasta, Ecoli.fasta, Leuco.fasta)を読み取り、その相補逆鎖（リバースコンプリメント）に変換して出力するプログラムを作成し、###Yamada09revcomp.c（###は学生番号の下3桁（過年度生は1文字目を9とし、残り二桁は学生番号の下2桁）Yamadaは自分の苗字で置き換え）というファイル名でメールのタイトルを「09基本-山田」（山田は自分の苗字で置き換え）などとして **knakamura.maebit@gmail.com**まで添付文書（OneDriveやGoogleDriveのファイルドロップ等にしない）で提出しなさい。

と言うファイル名を付してメールで提出しなさい。相補逆鎖とは、1) AをT, TをA, GをC, CをG に置き換え、2)順序を逆にしたものである。

例えば**ATGCGGG**と言う配列に対して 1) の相補をとる操作を施すと、**TACGCCC** となり、続いて配列を逆順として、**CCC GCAT** としたものが元の配列ATGCGGGのリバースコンプリメントとなっている。500万塩基ほどの非常に長い配列について行うためには、まず全配列をメモリ上に読み込み、続いて配列を後ろから対応する塩基に置き換えながら出力する。また、この時、**入力ファイルと同様に60塩基おきに改行を入れて見やすいファイルとする。**

以下、ソースコードの例を適宜参照

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

int main(int argc, char **argv)
{
    int i;
    char c;
    //////////////////////////////////////
    ////////// ここにこれまでの実習同様、入力ファイル名をargv[1] から取り、 //////////
    ////////// ファイルハンドルinfileとして開くコードを記述する //////////
    //////////////////////////////////////

    FILE *outfile;                // 出力ファイルハンドル
    char outfilename[132];         // 出力ファイル名
    sprintf(outfilename,"%s.out.fasta",argv[1]); // 出力ファイル名は入力ファイル名の後にout.fastaつけたもの
    outfile = fopen(outfilename,"w"); // 出力ファイルをオープン

    int num_base = 0;              // 塩基数
    int num_line = 0;              // 行数 (タイトル行以外)
```

```

char title[512];
fgets(title,512,infile);

// まず、タイトル行を1行だけ、分けて読み込んでおく

////////////////////////////////一度目のループでは塩基数（文字数）をカウントするだけ
while((c=fgetc(infile)) != EOF)
{
    if(c == '\n')
        num_line ++;
    else
        num_base ++;
}
// 塩基数カウント この数の文字を記憶する領域をあとで確保する

////////////////////////////////カウントした文字数分の記憶領域を確保 (malloc)
char *bases;
bases = (char *)malloc((int)(sizeof(char)*(num_base+2)));
// 文字配列の先頭アドレスを指すポインタを設定
// 一度目のスキャンでカウントした長さの文字列を確保
// '\0'の場所などすこし大きめに +2 として確保
// メモリの確保に失敗したらエラーメッセージを出して終了

if(bases == NULL)
{
    printf("Failed to allocate memory for sequence.\n");
    return 1;
}

rewind(infile);
fgets(title,512,infile);
// ファイルハンドルをファイルの先頭に巻き戻す
// タイトル行を読み飛ばす

////////////////////////////////二度目のループで確保した領域にデータを格納していく
int counter = 0;
while((c=fgetc(infile)) != EOF)
{
    if(c != '\n')
    {
        bases[counter] = c;
        counter ++;
    }
}
bases[counter] = '\0'
// 文字列の最後にNULL文字を詰める

////////////////////////////////これでファイルの文字列がすべてメモリ上に乗る
// 今日の課題では、ここから、まずタイトル行を1行書き出し、文字列を一字ずつ出力する、また60文字置きに改行文字 '\n' を出力する。

fprintf(outfile,"%s:RevComp sequence",title);
for(i=0;i<num_base;i++)
{
    // 元のタイトルに :RevComp sequence を付加して出力
    // 塩基数のループ 違う書き方でも良い

    //////////////////////////////////
    // リバースコンプリメントとなる文字列を一字ずつ出力するコードをここに記述する //
    //////////////////////////////////
    if( //ここに改行する条件を書く// ) // 60文字置きに改行
    {
        fprintf(outfile,"\n");
    }
}

fprintf(outfile,"\n");
return 0;
// 最後に改行（ただしnum_baseが60の倍数だと改行しなくてよい）
// メイン関数終了

```

応用課題1 プログラム名 mfrc(multi-fasta reverse compliment)（必須：提出期限、6月13日、できるだけ6月9日までをお願いします）

*Saccharomyces cerevisiae*は酵母、とよばれる真核生物である。そのゲノムは複数の染色体からなり、*Saccharomyces.fasta*（右下のQRコードのリンクにある）には複数の染色体の塩基配列が入っている。それぞれの染色体配列のはじめには、>で始まるタイトル行があり、このタイトル行のはじめ以外には > の文字は出現しない。これを読み込み、すでに作成したプログラム同様に、それぞれの染色体の相補逆鎖を出力するプログラムを作成し、  
**###Yamada09mfrcp.c**（###は学生番号の下3桁（過年度生は1文字目を9とし、残り二桁は学生番号の下2桁）Yamadaは自分の苗字で置き換え）というファイル名でメールのタイトルを「09応用-山田太郎」（山田太郎を自分の名前で置き換え）などとしてknakamura.maebit@gmail.comまで添付文書（OneDriveやGoogleDriveのファイルドロップ等にしない）で提出しなさい。