

先週の課題 `city.c` について、1.構造体の市の名前を20文字の決め打ちではなく、ファイルを読みながら必要な文字数分だけ確保する書き方に変更して下さい。

まず、ヘッダファイル`city.h`の中の構造体宣言中

```
typedef struct city_info
{
    char *name;           // 市の名前 ポインタ宣言でそれぞれの長さ分をファイルを見てから確保
    int  pop;             // 人口
    float area;           // 面積（平方キロメートル）
    float population_density; // 人口密度
} city_info;
```

市の名前の変数を

```
char* name;
```

とポインタ宣言します。

プログラムはメイン関数の中でまず

```
char  nbuff[64]; // 一時的な市の名前保管用の文字列
int    nlen;     // 市の名前の文字列の長さ
```

を定義しておいて、

読込の過程で、`while(fgets(buff,512,infile))` ループ内で1行ずつよみながら `int cc;` を何行目を読んでいるかのカウンターとすると

```
sscanf(buff,"%s %d %s",nbuff,cities[cc].pop,cities[cc].area);
```

のように、まず長めのバッファ変数`nbuff`に市名を読み込み

```
nlen = strlen(nbuff) + 1;
```

とします。`strlen`は `\0` をカウントしないので `1` を加えました。

ついで

```
cities[i].name = ( ) malloc ( );
```

で市名がはいる文字列のメモリを確保し、

```
strcpy(cities[i].name,nbuff);
```

としてコピーします

できたプログラムの名前を、`cities.c` とし、ヘッダファイル `city.h` とともに `"012yamada11cities"` というディレクトリに入れ、

```
tar -cf 012yamada11cities.tar 012yamada11cities
gzip 012yamada11cities.tar
```

により圧縮ファイルを作成し、できた圧縮ファイル `012yamada11cities.tar.gz` を、

中村_11基本 というタイトルを付けてメール添付で送信して下さい。012は自分の学生番号下3桁、yamadaおよび山田は自分の名前と置き換えるように！（ファイル／ディレクトリ名の名前はアルファベット）同姓の人がいる場合には、NakamuraKのようにfirst name の頭文字を姓の「あと」に付けて表記してください

応用課題11（任意：期限1週間）

先週の応用課題12について、出力ファイルが塩基配列ではなく、翻訳されたアミノ酸配列となるように改良する。塩基配列ははじめから3文字ずつ、一つのアミノ酸に置き換える。置き換えの規則はコドン表により決められている。塩基配列のATGCのうちAGCはそのまま、TはUに読み替えて下記のコドン表を参照する。コドン表の左上UUUはアミノ酸フェニルアラニン Phe/F であり、/の右側のFの1文字表記を用いる。UAA,UAG,UGAの2つはSTOPと記されているが、これは停止記号であり、*で置き換える。AUGは開始コドンを兼ねるが M と表記すれば良い。

たとえば

ATCGCCTTTTAAは、まず、AUCGCCUUUUAAと読み替えて、3文字ずつに区切って、AUC GCC UUU UAA、それぞれの3塩基をコドン表に従って I A F * となる

アミノ酸配列出力は塩基配列と同様60文字で改行する。

結果を確認する方法の一つは、すべてのcds配列で、アルファベットで表記されるアミノ酸が並んだ後に、最後にひとつだけ* = STOPが来るようになるはずである。またSTOPは配列の途中（最後以外）には現れない。

	U		C		A		G		
U	UUU	Phe/F	UCU	Ser/S	UAU	Tyr/Y	UGU	Cys/C	U
	UUC	Phe/F	UCC	Ser/S	UAC	Tyr/Y	UGC	Cys/C	C
	UUA	Leu/L	UCA	Ser/S	UAA	Stop	UGA	Stop	A
	UUG	Leu/L	UCG	Ser/S	UAG	Stop	UGG	Trp/W	G
C	CUU	Leu/L	CCU	Pro/P	CAU	His/H	CGU	Arg/R	U
	CUC	Leu/L	CCC	Pro/P	CAC	His/H	CGC	Arg/R	C
	CUA	Leu/L	CCA	Pro/P	CAA	Gln/Q	CGA	Arg/R	A
	CUG	Leu/L	CCG	Pro/P	CAG	Gln/Q	CGG	Arg/R	G
A	AUU	Ile/I	ACU	Thr/T	AAU	Asn/N	AGU	Ser/S	U
	AUC	Ile/I	ACC	Thr/T	AAC	Asn/N	AGC	Ser/S	C
	AUA	Ile/I	ACA	Thr/T	AAA	Lys/K	AGA	Arg/R	A
	AUG	Met/M (Start)	ACG	Thr/T	AAG	Lys/K	AGG	Arg/R	G
G	GUU	Val/V	GCU	Ala/A	GAU	Asp/D	GGU	Gly/G	U
	GUC	Val/V	GCC	Ala/A	GAC	Asp/D	GGC	Gly/G	C
	GUA	Val/V	GCA	Ala/A	GAA	Glu/E	GGA	Gly/G	A
	GUG	Val/V	GCG	Ala/A	GAG	Glu/E	GGG	Gly/G	G