情報処理演習
(12) 総まとめ

知能システム学　准教授
万 偉偉（ワン ウェイウェイ）

スケジュール

- 今日　総まとめ
- 1月29日　最終課題を配る

復習

- 変数の種類 char, int, float, double
- 進数、ビット、バイト
- printf関数 %d, %f, %c, %s, %x
  - %表示桁数.小数点以下の桁数f
- scanf関数 %d, %lf, &
- 四則演算 +, -, *, /, %
- 数学関数 #include <math.h>
- マクロ定義　大文字　#define PI 3.14

復習

- 条件分岐
- 比較演算子 ==, >=, <=, !=
- ブロック {}
- かつ、または &&, ||
- インデントと括弧でプログラムを綺麗にする

復習

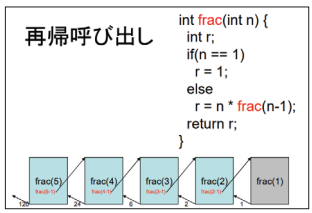
- 繰り返し
- for
  - for(初期化；繰り返しの条件；繰り返すたびに実行するもの)
- break即座に抜ける, continue次の処理へ進める, switch-case
- 計算の省略形 i++, i+=1
- i++と++iの違い

復習

- 繰り返し
- for
  - for(初期化；繰り返しの条件；繰り返すたびに実行するもの)
- break即座に抜ける, continue次の処理へ進める, switch-case
- 計算の省略形 i++, i+=1
- i++と++iの違い

復習

- 関数
  - プロトタイプの宣言
  - 戻り値　関数名（引数1，引数2，…）
- 標準ライブラリの関数stdio.h, math.h
- 再帰呼び出し



復習

- 配列
  - 同じ型のデータを並べたもの
  - データ型名　配列名[配列の要素数]
    - どんな型も良い、要素数は固定
  - 参照
    - 配列名[要素番号]、番号は0から
  - 二次元、三次元も可能
- 配列の初期化

復習

- 文字列
  - 文字:アスキーコード
  - 文字の配列　文字を並べたもの
  - 文字列　文字を並べたもの+ターミネータ
- getchar()
- cast 型を明示的に変換する
- 文字列操作する関数
  - strcat, strcmp, strncat,…

復習

- 制御文字 CR, LF, EOFなど

getchar()プログラムの入力

```
int mygetline(char s[], int size) {
        int c, i;
        for ( i = 0; (c = getchar()) != EOF &&
                        c != '¥n' && i < size - 1; i++ ) {
                s[i] = c;
        }
        s[i] = '¥0' ;
        if ( c == EOF && i == 0 ){
                return -1;
        }
        return i;
}
```

```
for(i=0; i<size-1; i++) {
        c = getchar();
        if(c==EOF or c=='¥n') {
                break;
        }
        s[i] = c;
}
```

復習

- 配列を引数に取る関数
  - 配列を受け取る関数であることを宣言
  - 関数の定義に要素数を指定する必要はない
  - 呼び出し側では、配列の名前のみ（添え字なし）
- const修飾子
- 二次元配列の引数渡し
  - 少なくとも二番名の引数の個数を指定すべき

復習

- メモリについて
  - メモリは複数のプログラムに共有されている
  - 1つのプログラムに必要なメモリを確保しないといけない
    - コードに従って、メモリのサイズを分かるはずです
    - この授業の範囲外の話
      - サイズがわからない場合
        - » 動的割り当てになる （この授業であれば、適当に多めに用意しても構いません）
      - OSから動的割り当てのため一部のメモリを特別用意している→ヒープメモリ

復習

- ポインタ
- 変数の位置は割られた単元のアドレス
  - データ型　＊変数名;
- この位置は以下の特徴を持っている.
  - 値の範囲は有効のアドレス.
  - データ型は割当たれた単元のサイズを示す.
- ポインタの演算
- C言語配列の謎、2重ポインタ

復習

- テキストファイル入出力
- FILE型変数　本質（来年の内容となり）
- FILE *fp_in=fopen("sample_data.txt","r");
- FILE *fp_out=fp_out = fopen(str, "a");
- fscanf, fprintfの活用
- 特に、書き出しのオプションについては書き出す ″w″ 以外にも，既存のファイルの最後に追加して書き出す ″a″ を使用することもできる.

実例　あみだくじ

あみだくじをしよう

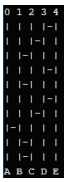
- 必要なもの
  - 縦線の数
  - 横線の数
    - 横線について
      - 隣り合う2本の縦線のみを水平につなぐ
    - ある高さにひける横線は1本だけとする

あみだくじをしよう

- 方針
  - 横線の位置は乱数で決め、あみだくじを生成する
  - あみだくじを表示する
  - あみだくじをする
  - あみだくじの結果を表示する
  - 結果は積算して最後に表示するようにして、繰り返す

ヒント

- 横線の位置を上から順に
  - srand(0) 乱数生成器を初期化
  - rand() で得られた乱数を縦線の数-1で割ってあまりで横線の位置を決める
  - 例



で、結果は

	0	1	2	3	4
A	0	1	0	0	0
B	1	0	0	0	0
C	0	0	1	0	0
D	0	0	0	0	1
E	0	0	0	1	0

研究しましょう

- どこから始めても、同じ確率で当たるのか
- 縦線の数と横線の数を一定して、違う乱数で当たる確率は同じですか
- 横線の数も乱数したらどうですか

終わりに

- これから、色んな言語を勉強したり、使ったりするかもしれませんが、C・C++言語必ず使われる．早くなれるのは専門として大事なこと
  - どのコースでも一緒
  - 一年生の段階で早めに授業を受けること
- 言語の勉強について暗記ではなく、実践が重要→座学の代わりに演習
- 一年生の内容は足りない気がしている方、ぜひ聞いてください（OSを作りたい、コンパイラーを作りたい）．
- 語法以外、言語に依存しないプログラミングの思想が重要→現存のパソコンアーキテクチャへ適用するアルゴリズムの考え方→知能システムコース二年のコンピュータ基礎（データ構造とアルゴリズム）