情報処理演習
(9)配列その2
文字列　続き

知能システム学　准教授
万 偉偉（ﾜﾝ ｳｪｲｳｪｲ）

課題の問題

- 戻り値について
  - 整数・小数などの戻り値がありましたら
    - 型名　関数名
    - return　変数名　する
  - 配列の戻り値がありましたら
    - void 関数名
    - 配列を引数として渡して　引用渡し
    - 関数内で配列の中身を変更する
    - returnをしない
  - 戻り値要らないなら　void

課題の問題

- 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
      - 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

課題の問題

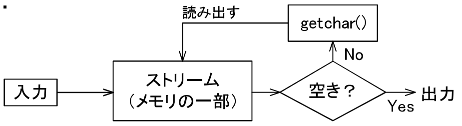
- 変数名・関数名の黙然なルール
  - 変数名　小文字
  - 関数名　小文字・小文字_小文字_小文字
  - define 全部大文字
- C++などの言語
  - クラス名　最初の文字は大文字
  - メンバ関数　小文字単語大文字からの単語
  - など

文字列

- アスキーコード
- 8bitの配列プラス終端記号
- 文字列の初期化
  - char s[] = "ABC";
    char s[] = { 'A', 'B', 'C', '\0' };
- 文字列の出力

getchar()プログラムの入力

- getchar()が実行されると，キーボードの入力を待つ．
- キーボードからの入力はリターンキーを押すまでは，プログラムに伝わらない．
- 毎回の入力は少なくても\nを含めます．
  - aとリターンなら -> aと\n二つの文字を得る
  - リターンだけなら -> \nを得る
- キーボードの入力はストリーム（メモリ）に保存される．getchar()は，入力ストリーム中の一つの文字を読み込む．



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;
}
```

文字列操作関数

- 文字列を取り扱う関数が用意されている
  - str・・・ という名称の関数群
  - strcat : 2つの文字列を接続する
  - strcmp : 文字列を比較する
    - 同じなら 0，違う場合は辞書順に -差 または +差
  - strcpy : 文字列をコピーする
  - strlen : 文字列の長さを返す
- 文字列長に制限のあるバージョンもある
  - strncmp, strncat など

文字列操作関数

```
#include <stdio.h>
#include <string.h>
int main(){
        char a[15] = "helloa";
        char b[15] = "hellob";
        char c[20];
        strcat(a, b);
        printf("%s", a);
        printf("\n");
        printf("%d", strcmp(a, b));
        printf("\n");
        printf("%d", strncmp(a, b, 6));
        printf("\n");
        strcpy(c, b);
        printf("%s", c);
        printf("\n");
        printf("%d", strlen(c));
        return 0;
}
```

CRとLF

- パソコンによって，改行のコードは違う
  - Windows : CR LF (0d 0a)
  - Unix : LF 0a
  - Mac : CR 0d
- CRとLFとは？
  - CR: Carriage Return（行頭に戻る ）
  - LF: Line Feed (行を送る)

元はラインプリンタのヘッドの制御から来ている
ラインプリンタでは，ヘッドを行頭に戻してから，行を送っていた

CRとLF

```
int main() {
        printf("Hello world");
        printf("\n");
        printf("hello japan");
        printf("%c", 0x0d);
        printf("hello ou");
        return 0;
}
```

```
int main() {
        printf("Hello world");
        printf("\n");
        printf("hello japan");
        printf("%c", 0x0d);
        printf("%c", 0x0a);
        printf("hello ou");
        return 0;
}
```