9 PAGES

課題 3

アルゴリズムとデータ構造B

第09回

本日の予定

課題３

- レポート提出期限：**2024年11月5日（水）9:15**

スタックのプログラミング

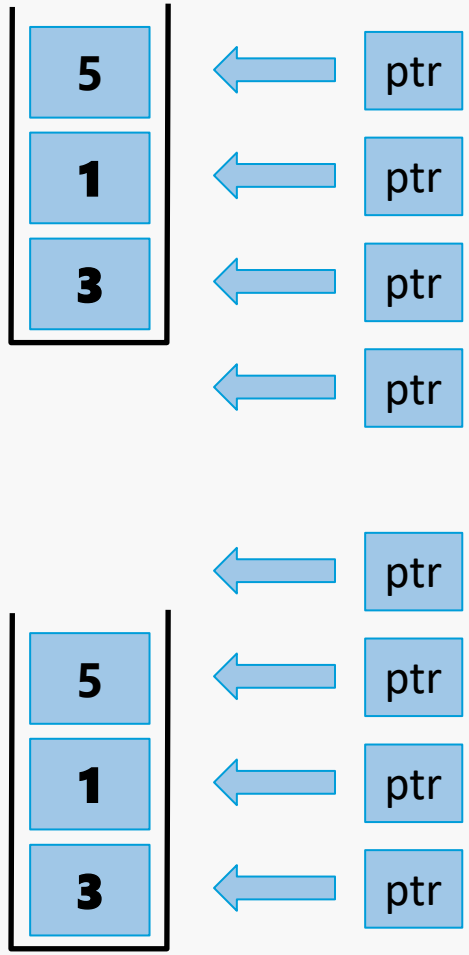
データ構造

- 配列　`int stk[MAX];`
- スタックポインタ　`int ptr =-1 ;`

} 簡単のため グローバル変数 (要改善)

アルゴリズム

- データの入力　`int Push(int x);`
  - ✓ スタックポインタの位置にデータを格納する
  - ✓ スタックポインタを1つずらす（進める）
- データの出力　`int Pop();`
  - ✓ スタックポインタの位置からデータを取り出す
  - ✓ スタックポインタを1つずらす（戻す）



ずらしてから入れる（出す）のか、入れ（出し）てからずらすのか？
⇒ ptr の初期値によってアルゴリズムが異なる

Push のアルゴリズムの確認


```
int Push(int x) {
    if (ptr < MAX-1)
        stk[++ptr] = x;
    else
        return -1;

    return 0;
}
```

方針：ずらしてから入れる
スタックが full のときは -1 を返す

Pop のアルゴリズムの確認


```
int Pop() {
    if (ptr >= 0)
        return stk[ptr--];
    else
        return -1;

}
```

方針：出してからずらす
スタックがemptyのときは -1 を返す

ex07.c Display()関数

- 回答例

```c
// 設問４：Display()の完成
void Display() {
    int i;
    for (i = MAX-1; i ≥ 0; i--) { // スタックの上から下に for文を回す
        if (i == ptr) // スタックポインタの位置のとき PTR→ を出力
            printf("PTR→");
        else
            printf("     ");
        // 要素番号とスタックの値を出力
        printf("%6d: %6d\n", i, stk[i]);
    }
}
```

課題３

指示に従ってプログラムを作成し，
期限までにレポートを Teams 上で提出すること

- レポート提出期限：2025年11月5日（水）9:15

逆ポーランド記法

- 逆ポーランド記法：数式の後置記法
  - ✓ a + b ⇒ a b +
  - ✓ 普通の数式は中置記法
  - ✓ 後置記法は，コンパイラで利用される
    - 計算アルゴリズムが単純，（）も必要ない

後の授業で中値記法から逆ポーランド記法への変換アルゴリズムを学習します

課題３　ファイルからの読み込み

- 前提：ファイルには空白区切りで，数値，演算子，式の終わりを示す記号 “end” が記入
  ⇨ **文字列として読み込めば良い（文字列は空白で自動的に区切られる）**
- fpでファイルオープン，char c[10];に読み込むとして，
  - ✓ fscanf() を用いて：fscanf(fp, “%s”, c);
- ファイルの終端は **EOF** で表される．よってファイル終端まで読み込むためには
  - ✓ while(fscanf(fp, “%s”, c) != **EOF**){ ...

とすれば良い．while の中 {} で c に代入された
文字列（数値 or 演算子 or 式の終わりの記号）を処理可能

課題３　数値と演算子の扱い

- 読み込んだ文字列が数値であるか，演算子であるか，
  式の終わりの記号 “end” であるかを区別する必要.
- 文字列を比較する関数（課題１で使用したもの）で場合分けすると良い
- 文字列が数値であった場合，スタックへ PUSH するが，
  int 型を扱うスタックとして実装しているため，
  文字列 → 整数の変換　が必要
  - ✓ `// char c[10]` に “24” がセットされているとして
    `int num = atoi(c);  // num 24 がセットされる`