情報構造論 (斉藤)　後期中間試験　　番号：

4EI　2019/12/2　(1/3)　　氏名：

以下の 6 つの設問の中から、5 つを選んで解答せよ。

1　スタックマシンのトレース

```
int stack[ 10 ] ;   |  void push( int x ) { *sp++ = x ; }
int* sp = stack ;   |  int pop()          { return *--sp ; }
                    +-----------------------------------------
void eval( char* p ) {
   for( ; *p != '\0' ; p++ ) {
      if ( '0' <= *p && *p <= '9' )
         push( *p - '0' ) ;
      else
         switch( *p ) {
         case '+' : push( pop() + pop() ) ;
                    break ;
         case '-' : push( -pop() ) ;
                    break ;
         }
   }
}
```

(設問)

このプログラムの実行結果を答えよ (10x2)
ただし stack の変化が判るような説明を
つけること。

```
                    +-----------------------------------------
                    |  void main() {
                    |     printf( "%d\n" , eval( "321++" )    ) ;
                    |     printf( "%d\n" , eval( "43+21+-+" ) ) ;
                    |  }
```

2　BNF 記法

(1) 次の BNF 記法で定義される<変数名>に (A)〜(E) の表現が、合致する○/合致しない×で答えよ。(3x4)

```
<数字> ::= 0|1|2|3|4|5|6|7|8|9 ;
<英字> ::= A|B|C|D|E|F ;
<英数字> ::= <英字> | <数字> | _ ;
<変数名> ::= <英字> | <変数名><英数字> ;
```

(A) A_8_A　[______] (B) 246　[______] (C) 3E5　[______] (D) F5_1　[______]

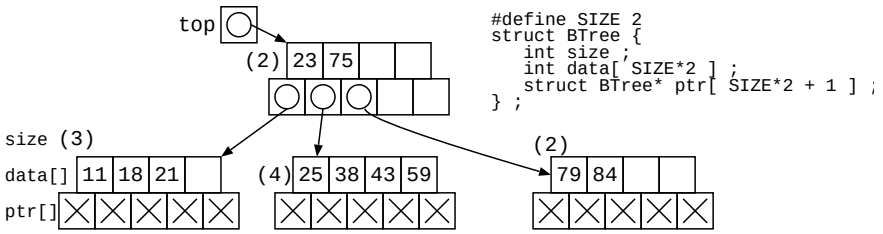
(2) 以下の下線部 (E),(F) にふさわしい処理プログラムの名前を答えよ。(4x2)

C 言語のソースコードので始まる行は、(E)_____で#include, #define などの変換が行ったあと、コンパイラが中間コードに変換する。その後、(F)_____によって他の中間コードやライブラリと結合され、実行できる機械語が作られる。

3　B 木の処理

以下の図に示されるように B 木のデータ構造を定義してある。

この木構造のポインタ p と、探すデータ key を渡し、データが見つかったら 1, 見つかなければ 0 を返す find_btree() を完成せよ。




```
      find_btree(                    p ,       key ) {
~~~~~(A)2         ~~~~~~~~~~~~~~~~~~(B)3 ~~~~~(C)3
   if ( ~~~~~~~~~~~~~ ) {
        ~~~~~~~~~~~~~(D)4
      return 0 ; // 見つからなかった
   } else {
      int i ;
      for( i = 0 ;                   ; i++ ) {
                    ~~~~~~~~~~~~~~~~~(E)4
         if ( key < p->data[ i ] )
            break ;
         else if (                        )
                   ~~~~~~~~~~~~~~~~~~~~~~(F)4
            return 1 ;  // 見つかった
      }
      return find_btree( p->ptr[ i ] , key ) ;
   }
}
void main() {
   if ( find_btree( top , 21 ) )
      printf( "みつかった\n" ) ;
}
```

4　説明問題

以下の 2 つのデータ構造について、イメージ図を交えながら説明せよ。

1. AVL 木について、AVL を使わない場合の問題点などを交えながら、具体的に説明せよ。(10)
2. 番兵付きの双方向リストについて、データ構造の宣言を示した上で、具体的に説明せよ。(10)

5　文字列の 2 分木

文字列の

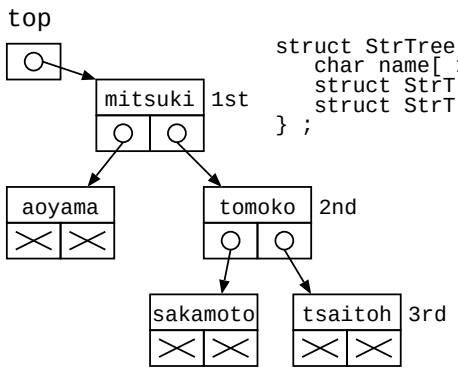
```
struct StrTree* cons_StrTree(
         char*          n ,
         struct StrTree* L ,
         struct StrTree* R ) {
   struct StrTree* ans ;
   ans = (struct StrTree*)malloc(
              sizeof( struct StrTree ) ) ;
              ~~~~~~~~~~~~~~~~~~~~~~(A)
   if ( ans != NULL ) {
      strcpy( ans->name , n ) ;
              ~~~~~~~~~(B)
      ans->left  = L ;
      ans->right = R ;
   }
   return ans ;
}
struct StrTree* top = NULL ;

void insert_StrTree( __________ s ) {
                     ~~~~~~~~~~(C)2
   struct StrTree** tail = ________ ;
                           ~~~~~~~~(D)3
   while( *tail != NULL ) {

      int ans = strcmp( (*tail)->name , s ) ;

      if ( ____________ )
           ~~~~~~~~~~~~(E)2
         break ;
      else if ( ans > 0 )
                ~~~~~~~(F)
         tail = ___________________________ ;
      else      ~~~~~~~~~~~~~~~~~~~~~~~~~(G)3
         tail = &( (*tail)->right ) ;
                ~~~~~~~~~~~~~~~~~~(H)
   }
   if ( *tail == NULL )
      *tail = cons_StrTree( s , NULL , NULL ) ;
              ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~(I)
}

void main() {
  char string[ 100 ] ;
  while( scanf( "%s" , string ) == 1 ) {
                ~~~~(J)
    insert_StrTree( string ) ;
  }
  print_StrTree( top ) ;
}
```




```
struct StrTree {
   char name[ 10 ] ;
   struct StrTree* left ;
   struct StrTree* right ;
} ;
```

下線部の型を答えよ (2x5)

(A)_____

(B)_____

(F)_____

(H)_____

(I)_____

(C),(D),(E),(G) に
適切な処理を答えよ
下線部に記入せよ。

6　2 分木の深さ

前設問のような StrTree において、木構造の最大段数を求める関数を作成せよ。NULL なら 0 段、前設問の図では 3 段という結果が欲しい。

```
int depth( struct StrTree* p ) {
```