情報通信工学実験 IB 第 6 週課題

学籍番号: 氏名:

連結リストのセルを表す構造体を記述せよ.

情報通信工学実験 IB 第 6 週課題

学籍番号: 氏名:

二分木の節点を表す構造体を記述せよ.

情報通信工学実験 IB 第 6 週課題

学籍番号: 氏名:

下記に示す構造体で表される連結リストが次のとおり記憶されているとする. このとき, 変数 cell1 を用いて cell3->data の値を 10 に変更せよ.

```
struct cell {
    int data;
    struct cell *next;
};

struct cell *cell1, *cell2, *cell3;

cell1->data = 2;      cell2->data = 7;      cell3->data = 3;
cell1->next = cell2;  cell2->next = cell3;  cell3->next = NULL;
```

情報通信工学実験 IB 第 6 週課題

学籍番号: 氏名:

下記に示す構造体で表される二分木が次のとおり記憶されているとする．このとき, 変数 node1 を用いて node4->data の値を 10 に変更せよ.

```
struct node {
     int data;
     struct node *left;
     struct node *right;
 };

 struct node *node1, *node2, *node3, *node4, *node5;

 node1->data = 2;      node2->data = 7;      node3->data = 3;
 node1->left = node2;  node2->left = node5;  node3->left = NULL;
 node1->right = node3; node2->right = NULL;  node3->right = node4;
```

情報通信工学実験 IB 第 6 週課題

学籍番号: 氏名:

フィボナッチ数列を求める次の再帰関数の *A* を埋めよ.

```
int fibonacci(int num)
 {
   int result;  /* calculation result */

   if (num <= 2) {
     result = 1;
   } else {
     *A*
   }

   return result;
 }
```

情報通信工学実験 IB 第 6 週課題

学籍番号: 氏名:

ある正の整数値の階乗を求める次の再帰関数の *A* を埋めよ.

```
int factorial(int num)
 {
   int result;  /* calculation result */

   if (num == 1) {
     result = 1;
   } else {
     *A*
   }

   return result;
 }
```

情報通信工学実験 IB 第 6 週課題

学籍番号: 氏名:

スタックのトップセルが保持しているデータを取得する関数 `int topStack(struct cell *stack);` をリストの操作関数群を使って記述せよ. ただし, スタックの底はリストの先頭セルとする.

リストの操作関数群:

```
struct cell *makeNullList(void);
struct cell *nextCell(struct cell *target, struct cell *list);
struct cell *firstCell(struct cell *list);
struct cell *endCell(struct cell *list);
struct cell *previousCell(struct cell *target, struct cell *list);
struct cell *insertCell(int data, struct cell *targt, struct cell *list);
struct cell *deleteCell(struct cell *target, struct cell *list);
int retrieveCell(struct cell *target, struct cell *list);
struct cell *locate(int data, struct cell *list);
```

情報通信工学実験 IB 第 6 週課題

学籍番号: 氏名:

キューの先頭データを取得する関数 `int frontQueue(struct cell *queue);` をリストの操作関数群を使って記述せよ. ただし, キューの終端はリストの先頭セルとする.

リストの操作関数群:

```
struct cell *makeNullList(void);
struct cell *nextCell(struct cell *target, struct cell *list);
struct cell *firstCell(struct cell *list);
struct cell *endCell(struct cell *list);
struct cell *previousCell(struct cell *target, struct cell *list);
struct cell *insertCell(int data, struct cell *targt, struct cell *list);
struct cell *deleteCell(struct cell *target, struct cell *list);
int retrieveCell(struct cell *target, struct cell *list);
struct cell *locate(int data, struct cell *list);
```

情報通信工学実験 IB 第 6 週課題

学籍番号: 氏名:

リストを逆順に並べ替える関数 struct cell *reverseList(struct cell *list); の *A* を埋めよ.

リストの操作関数群:

```
struct cell *makeNullList(void);
struct cell *nextCell(struct cell *target, struct cell *list);
struct cell *firstCell(struct cell *list);
struct cell *endCell(struct cell *list);
struct cell *previousCell(struct cell *target, struct cell *list);
struct cell *insertCell(int data, struct cell *targt, struct cell *list);
struct cell *deleteCell(struct cell *target, struct cell *list);
int retrieveCell(struct cell *target, struct cell *list);
struct cell *locate(int data, struct cell *list);
```

```
struct cell *reverseList(struct cell *list)
{
  struct cell *rev;  /* reversed list */
  int data;  /* cell data */

  if (firstCell(list) != (struct cell*)NULL) {
    /* retrieve and delete first cell */
    data = retrieveCell(firstCell(list), list);
    deleteCell(firstCell(list), list);

    /* reverse list */
    rev = reverseList(list);

    /* insert retrieved data to tail */
    *A*
  } else {
    rev = makeNullList();
  }

  return rev;
}
```

情報通信工学実験 IB 第 6 週課題

学籍番号: 氏名:

次のプログラムを見て, 間違いがあればそれを修正せよ.

```
struct cell {
  int data;
  struct cell *next;
};

void func()
{
  struct cell *cell1, *cell2;

  cell1->data = 10;
  cell1->next = (struct cell*)NULL;

  cell2 = cell1;

  printf("cell2->data = %d\n", cell2->data);

  free(cell1);
}
```

情報通信工学実験 IB 第 6 週課題

学籍番号: 氏名:

二分木を通りがけ順に表示する関数 `void inOrder(struct node *tree);` の `*A*` および `*B*` を埋めよ.

```
void inOrder(struct node *tree)
{
  if (tree != (struct node*)NULL) {
    if(tree->left != (struct node*)NULL) {
      *A*
    }

    printf("%d, ", tree->data);

    if(tree->right != (struct node*)NULL) {
      *B*
    }
  }
}
```

情報通信工学実験 IB 第 6 週課題

学籍番号: 氏名:

実行ファイル `test` を作成するためには，C のソースファイル `test.c` をコンパイルおよびリンクする必要があるとき，実行ファイル `test` を作成するための `Makefile` を記述せよ．

情報通信工学実験 IB 第 6 週課題

学籍番号: 氏名:

分割コンパイルを行うとき，ヘッダファイルを別に作成することがあるが，作成する関数のプロトタイプ宣言をヘッダファイル内に記述する意味を述べよ．

情報通信工学実験 IB 第 6 週課題

学籍番号: 氏名:

次のプログラムを実行した際，表示されるものは何か．その理由とともに述べよ．

```
struct st {
  int data;
  struct st *next;
};

int main(void)
{
  struct st stA, stB, stC;

  stA.data = 12;
  stA.next = &stC;
  stB.data = 35;
  stB.next = &stA;
  stC.data = 27;
  stC.next = &stB;

  printf("%d\n", (stB.next)->data);

  return 0;
}
```

情報通信工学実験 IB 第 6 週課題

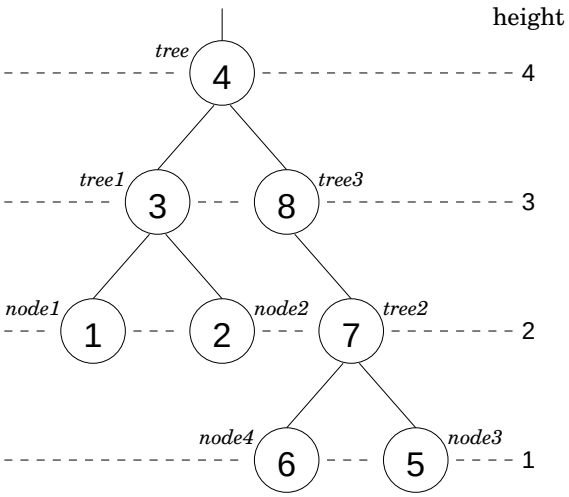
学籍番号: 氏名:

図で示される二分木が次の関数の引数として与えられたとき，表示されるものは何か．その理由とともに述べよ．

```
struct node {
  int data;
  struct node *left;
  struct node *right;
};

int func(struct node *tree)
{
  struct node *p;

  if (tree->left != (struct node*)NULL) {
    func(tree->left);
  } else {
    printf("%d\n", tree->data);
  }
}
```



情報通信工学実験 IB 第 6 週課題

学籍番号: 氏名: