

以下の6つの設問の中から、5つを選んで回答せよ。

## 1 基本動作トレース (20)

```

struct List {
    int data ;
    struct List* next ;
} ;

struct List* cons( int x , struct List* n ) {
    struct List* ans ;
    ans = (struct List*)malloc( sizeof( struct List ) ) ;
    if ( ans != NULL ) {
        ans->data = x ;
        ans->next = n ;
    }
    return ans ;
}

void main() {
    struct List* a
        = cons( 2 , cons( 3 ,
            cons( 4 , NULL ) ) ) ;
    struct List* b
        = even_filter( a ) ;
}

struct List* even_filter( struct List* p ) {
    struct List* ans = NULL ;
    struct List** tail = &ans ;
    for( ; p != NULL ; p = p->next ) {
        if ( p->data % 2 == 0 ) {
            *tail = cons( p->data , NULL ) ;
            tail = &( (*tail)->next ) ;
        }
    }
    return ans ;
}

```

このプログラムを動かした時に関数 main で生成されたリスト a, b のデータ構造のイメージ図を解答し、またプログラム中の (A)-(F) の型を答えよ。

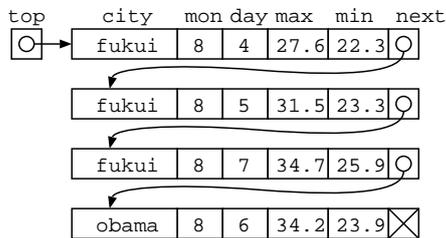
[イメージ図]

a  
b

[型の解答欄] (4+4+2x6)

(A) \_\_\_\_\_ (D) \_\_\_\_\_  
 (B) \_\_\_\_\_ (E) \_\_\_\_\_  
 (C) \_\_\_\_\_ (F) \_\_\_\_\_

## 2 プログラム記述問題 (20)



左図のような市町村名, 月, 日, 最高気温, 最低気温の情報がリスト構造で与えられるものとする。

- この情報を保存するための、構造体の宣言、
- この構造体の先頭アドレス top が引数で与えられた場合、真夏日 (最高気温が 30 度以上) でかつ熱帯夜 (最低気温が 25 度以上) の日の市町村名と日付を表示する関数 print\_heavy\_hot\_day() を作成せよ。

### 3 穴埋め問題 (20)

```
void print( struct List* p ) {
    for( ; p != NULL ; p = p->next )
        printf( "%d " , p->data ) ;
    printf( "\n" ) ;
}

void main() {
    struct List* a
        = cons( 2 , cons( 3 ,
            cons( 2 , NULL ) ) ) ;
    print( uniq( a ) ) ; // output=3,2

    uniq( p ) {
        struct List* ans = NULL ;
        for( ; ) {
            if ( )
                ans = cons( p->data , ans ) ;
        }
        return ans ;
    }
}
```

```
int find( p , key ) {
    for( ; p != NULL ; ) {
        if ( p->data == key )
            return 1 ;
    }
    return 0 ;
}
```

リスト構造で集合のデータを扱うプログラムを作成する。  
A = {2,3,2} という集合では同じデータ 2 が含まれるので同じ値が 1 つしか含まれないリストを生成する。uniq() を完成させる。  
下線部 (A) ~ (H) を埋めよ。

List, cons() は [設問 1] を継続

### 4 動作トレースと説明問題 (20)

```
struct List* stack = NULL ;
void push( int x ) {
    stack = cons( x , stack ) ;
}
int pop() {
    int ans = stack->data ;
    struct List* d = stack ;
    stack = stack->next ;
    free( d ) ;
    return ans ;
}

void add() {
    int r = pop() , l = pop() ;
    push( l + r ) ;
}
void mul() {
    int r = pop() , l = pop() ;
    push( l * r ) ;
}
void main() {
    push( 1 ) ; push( 2 ) ; add() ;
    push( 3 ) ; mul() ;
    printf( "%d\n" , pop() ) ;
}
```

(5+5+10)

- 上記のプログラムの実行結果を答えよ。 List, cons() は [設問 1] の継続とする。
- このプログラムにおけるスタックのデータの出し入れの特徴を 4 単語で表せ。
- 配列を用いた「待ち行列」の実装方法について、図などを交えながら説明せよ。

## 5 説明問題 (20)

下線部 (A) ~ (X) を穴埋めせよ。回答欄 (B), (C), (D), (E) はオーダー記法で回答すること。

1. リスト構造と配列を比べた場合、配列の中から目的のデータを探すのにかかる時間は、あらかじめデータを昇順に並べておく (A) \_\_\_\_\_法を用いれば、検索に要する時間は、データ件数を  $N$  としたとき (B) \_\_\_\_\_となる。しかしリスト構造であれば、先頭から1つずつ探すことから処理時間は、(C) \_\_\_\_\_となる。(3+2+2)
2. 一方で、配列では昇順のデータ列の途中にデータを挿入する場合、データを挿入すべき場所を見つけた後に、データを挿入する場所を確保するためにデータを1つずらす処理が必要となる。この処理には (D) \_\_\_\_\_の時間がかかる。これに対して、リスト構造であれば、データの挿入場所を見つけた後の挿入には (E) \_\_\_\_\_の時間がかかる。(2+2)
3. 最大でもデータ件数が200件、平均約100件の整数型のデータを保存する場合、配列であれば、このデータを覚えるのにメモリは (F) \_\_\_\_\_byte 必要となる。リスト構造の場合は `sizeof( struct List )` は、1件あたり (G) \_\_\_\_\_byte なので、このデータ全体を覚えるには (H) \_\_\_\_\_byte を必要とする。(3+3+3)

ただし、`sizeof(int)` は4とし、OSは64bitとする。ヒープ領域の管理に必要なメモリについては考慮しなくていい。

## 6 エラトステネスのふるい (20)

エラトステネスのふるいのアルゴリズムで、64未満の素数を表示するプログラムを以下の様に作成した。エラトステネスのアルゴリズムは、 $i$ の倍数に目印をつけていき、目印がついていないものを素数とする。

このプログラムでは、偶数は2以外は素数ではないので、奇数だけをbit列に保存する。そこで、奇数  $i$  は  $\frac{i-1}{2}$  番目のbitに保存する。

```
void main() { // ...11,9,7,5,3,1 i 番目
    int primes = 0 ; // ...0,1,0,0,0,0 ビット列
    printf( "2\n" ) ; // 2は偶数の素数

    for( int i = 3 ; _____ ; i += 2 ) { // 奇数だけの繰り返し
        if ( (primes & _____) == 0 ) {
            printf( "%d\n" , i ) ;
            for( int j = i + i ; _____ ; _____ ) {
                if ( _____ ) // jが奇数の時
                    primes = primes _____ ;
            }
        }
    }
}
```