

6 つの選択問題より、5 つを選んで回答せよ。

1 メモリ管理 (20)

以下のプログラムの実行結果を答えよ。(解答は右余白に)

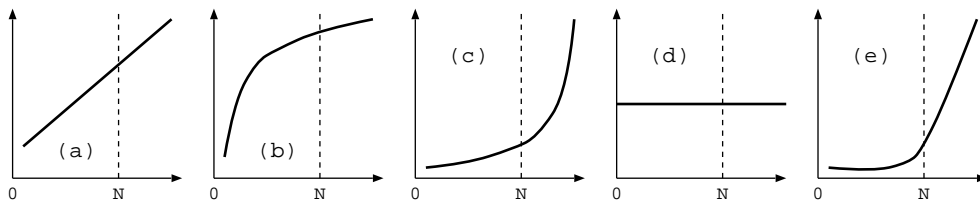
```

struct Data {
    int use ;
    int data ;
} array[ 5 ] = {
    {0,11}, {0,22}, {0,33},
    {0,44}, {0,55},
} ;
void gc() {
    int i ;
    free_top = -1 ;
    for( i = 0 ; i < 5 ; i++ )
        if ( !array[i].use ) {
            array[i].data = free_top ;
            free_top = i ;
        }
}
int free_top = -1 ;
void entry( int m , int x ) {
    array[ m ].data = x ;
    array[ m ].use = 1 ;
}
void main() {
    int i ;
    entry( 1 , 200 ) ;
    entry( 3 , 100 ) ;
    gc() ;
    for( p = 0 ; p < 5 ; p++ )
        printf( "%d,%d,%d\n" ,
            p ,
            array[p].use ,
            array[p].data ) ;
}
    
```

2 処理オーダの問題 (20)

以下に示す 4 つの処理の、データ件数に対する処理時間の変化のグラフにふさわしいものを (a) ~ (e) の中から、それぞれ選び、さらに (a) ~ (e) の中で、(1) ~ (4) に該当しなかったグラフを選び、そのような処理時間の変化をする処理の例を具体的にあげて説明せよ。

1. でたらめな順の配列の中から重複するデータを全て取り除く処理時間。
2. 双方向循環リストで、新たなデータを末尾に追加するための処理時間。
3. 電話番号を記憶している 2 分木に新たに 1 件の電話番号を追加する処理の時間。
4. 整数データを記憶している 2 分木で、全データの合計を求める処理時間。



(1) _____, (2) _____, (3) _____, (4) _____, (該当なし) _____

該当なしの処理例 (3x5+5)

3 説明問題 (10x2)

以下の2つについて具体的に説明せよ。

1. 参照カウンタ法について具体的に説明し、参照カウンタが活用されている事例を1つあげよ。
2. データ格納に、固定サイズの配列とリスト構造を用いた場合、それぞれの方式の利点を2つずつあげよ。

4 データ構造設計 (20)

世界の独立国には、国ドメインと呼ばれる英大文字2文字 (例:日本=JP) が割り当てられている。(ISO-3166) この国ドメインをキーとして、国 (国ドメイン, 正式国名, 国歌) を管理したい。

ただし、ISO-3166 の国ドメイン数 249カ国、漢字1字は char 型 2byte で扱い、以下の例を正しく記憶できること。

(例)

JP, 日本, 君が代

RU, ロシア連邦, ロシア連邦国歌

UK, グレートブリテン及び北部アイルランド連合王国, 神よ女王陛下を守り給え

国ドメインから、国の情報をすばやく検索するのにふさわしい (1) データ構造の宣言 を示し、新しい国が新たに1つ追加された場合、(2) その追加処理をどのように行うか を説明したうえで、国数 N 件の場合に (3) データ記憶に使用したメモリの総量 を答えよ。(7+7+6)

ただし、int, ポインタは1つあたり 4byte を使用し、malloc の呼び出しでは指定された byte 数+8byte を消費するものとする。また、国名の平均長さ L byte, 国歌名の平均長さ M byte とする。

5 ハッシュ法 (20)

```
#define SIZE 10
struct NameBirthday {
    char name[ 20 ] ;
    int mon , day ;
} hash[ SIZE ] ; //name[0]='\0'
int hash_func( char s[] ) {
    int sum = 0 , i ;
    for( i = 0 ; s[i] != '\0' ; i++ )
        sum += s[i] ;

    return _____ ;(A)3
}
void entry( char s[] , int m , int d ) {

    int i , idx = _____ ;(B)2
    for( i = 0 ; i < SIZE ; i++ ) {
        if ( hash[idx].name[0] == '\0' )
            break ;
        idx = (idx + 1) % SIZE ;
    }
    if ( hash[idx].name[0] == '\0' ) {
        strcpy( hash[ idx ].name , s ) ;
        _____(C)2
    } _____(D)2
}

[設問] 名前と誕生日のデータについて
オープンアドレス法によるプログラムの
下線部に相応しい処理を記述せよ。
-----
int find( char s[] ) {
    int i , idx = _____ ;(E)2
    for( i = 0 ; i < SIZE ; i++ ) {
        if ( _____ ) (F)3
            return -1 ;
        if ( strcmp( _____ ) == 0 ) (G)3
            return idx ;
        idx = _____(H)3
    }
}
void main() {
    entry( "t-saitoh" , 2 , 7 ) ;
    entry( "ayuka" , 9 , 18 ) ;
    entry( "mitsuki" , 7 , 14 ) ;
    printf( "%d\n" , find( "ayuka" ) ) ; // 0
    printf( "%d\n" , find( "tomoko" ) ) ; // -1
}
```

6 オブジェクト指向関連 (20)

以下のプログラムの実行結果を答え、(a)~(c)の用語について、簡単に説明せよ。

```
class A {
private:
    char name[10] ;
    int age ;
public:
    A( char s[] , int x ) {
        strcpy( name , s ) ;
        age = x ;
    }
    void print() {
        printf( "%s(%d)\n" ,
            name , age ) ;
    }
} ;

class B_A : public A {
private:
    int mon , day ;
public:
    B_A( char s[] , int x , int m , int d )
        : A( s , x ) {
        mon = m , day = d ;
    }
}
void main() {
    A mitsuki( "mitsuki" , 14 ) ;
    mitsuki.print() ;
    B_A ayuka( "ayuka" , 12 , 9 , 18 ) ;
    ayuka.print() ;
}
```

実行結果 (4x2)

(a) コンストラクタ (4)

(b) 基底クラス (4)

(c) 継承 (4)