

6 つの設問の中から、5 つを選んで回答せよ。

1 基本問題 (3 × 6 + 2)

1. プログラムと計算処理に必要なデータを合わせて、(A) _____ という。
2. メモリを大量に消費するプログラムが動いていると、OS の (B) _____ 機能が使われるので処理速度が遅くなる。
3. 処理時間がデータ件数に依存せず一定時間の時、オーダーは (C) _____ である。
4. クイックソートの処理時間のオーダーは、(D) _____ である。
5. スタックは、データの出し入れの特徴を表す 4 語の英単語で (E) _____ と呼ぶ。
6. 一般的な 32bit コンピュータで、sizeof(char) の値は (F) _____ である。
7. 動的メモリ確保の alloca() 関数は、メモリの確保に失敗すると (G) _____ を返す。

2 ポインタと型の理解 (20)

<pre> struct Charactor { char name[10] ; char* alias[2] ; } forger[4] = { { "Loid" , { "SPY" , "Twilight" } } , { "Anya" , { "ESPER" , "007" } } , { "Yor" , { "ASSASSIN" , "Briar Rose" } } , { "Bond" , { "PRECOGNITION" , "008" } } , } ; int main() { printf("%s\n" , forger[2].name) ; printf("%s\n" , forger[0].alias[1]) ; printf("%c\n" , forger[1].name[1]) ; struct Charactor* p ; p = forger + 2 ; printf("%s\n" , p->alias[0]) ; printf("%s\n" , p[-1].name) ; printf("%s\n" , *(forger->alias)) ; printf("%c\n" , **(forger[1].alias)) ; printf("%d\n" , p->name[3]) ; printf("%s\n" , (p-1)->alias[1]+2) ; p++ ; printf("%c\n" , *((*p).name)+2) ; } </pre>	<p>このプログラムを実行したときに 表示される内容を答えよ。</p> <p>回答欄</p> <p>~~~~~(A)2</p> <p>~~~~~(B)2</p> <p>~~~~~(C)2</p> <p>~~~~~(D)2</p> <p>~~~~~(E)2</p> <p>~~~~~(F)2</p> <p>~~~~~(G)2</p> <p>~~~~~(H)2</p> <p>~~~~~(I)2</p> <p>~~~~~(J)2</p>
---	--

3 動作予測 (20)

以下のプログラムの実行結果を答えよ。
(再帰に応じて変数の変化が分かる図を交えてあれば、中間点なども考慮する)

```
void sort_min( int array[] , int start , int size ) {
    if ( start + 1 < size ) {
        int min = start , tmp ;
        for( int i = start + 1 ; i < size ; i++ )
            if ( array[ min ] > array[ i ] )
                min = i ;
        printf( "a[%d]:%d <=> a[%d]:%d\n" ,
                start , array[ start ] ,
                min , array[ min ] ) ;
        tmp = array[ min ] ;
        array[ min ] = array[ start ] ;
        array[ start ] = tmp ;
        sort_min( array , start + 1 , size ) ;
    }
}

int main() {
    int a[] = { 11 , 33 , 22 } ;
    sort_min( a , 0 , 3 ) ;
    for( int i = 0 ; i < 3 ; i++ )
        printf( "%d " , a[ i ] ) ;
    printf( "\n" ) ;
    return 0 ;
}
```

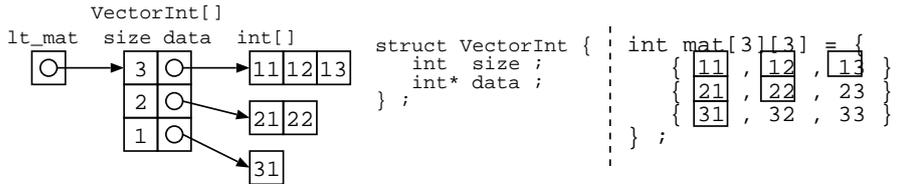
```
-----
(7)
-----
(7)
-----
(6)
```

4 オータ記法 (20)

- あるプログラムの処理時間が $T_1(N) = T_a + T_b \times N\sqrt{N} + T_c \times 2^N$ であった時、この処理時間のオーダーを答えよ。(5)
- ある処理プログラムのオーダーが $O(N^2)$ であったとき、 $N = 100$ 件で 1[msec] の時間がかかった。
 - $N = 1000$ 件での処理時間を予想せよ。(5)
 - 同じ処理を処理速度が3倍のコンピュータで実行すると、1[msec] で何件のデータを処理できるか予想せよ。(5)
- 設問3の `sort_min()` に示したような、末尾再帰のプログラムについて、特徴を説明せよ。(5)

5 穴埋め問題 (20)

2次元配列から、対角より下側の要素だけを取り出した上三角行列を作りたい。イメージ図を参考に、要素数と配列へのポインタからなる VectorInt を使った、以下のプログラムの下線部を埋めよ。



```
int main() {
    struct VectorInt* lt_mat ;
    lt_mat = (struct VectorInt*)malloc( _____ * 3 ) ;
    for( int i = 0 ; i < 3 ; i++ ) {
        lt_mat[ i ].size = _____ ;
        lt_mat[ i ].data = _____ ;
        for( int j = 0 ; j < lt_mat[ i ]. _____ ; j++ )
            _____ = mat[ i ][ j ] ;
    }
    // lt_mat を使った処理
    for( int i = 0 ; i < 3 ; i++ ) {
        free( _____ ) ;
    }
    _____ ;
    return 0 ;
}
```

6 プログラム作成問題 (20)

商品:(品番, 品名, 価格) のデータが与えられたら、品番購入数:(品番, 個数) を組み合わせて合計金額を求めたい。

- 商品のデータを保存するための構造体を宣言し、(6)
- 構造体の配列に、入力された商品情報を読み込み、(8)
ただしデータ件数は最初に入力として与えられる。
- buy_table[] の品番購入数を元に合計金額を出力せよ。(6)

```
struct Buy {
    int id ;
    int count ;
} buy_table[] = {
    { 1010 , 5 } ,
    { 1022 , 1 } ,
    { 1020 , 2 } ,
};
```

| 入力
 | 3
 | 1010 Orange 50
 | 1020 Apple 100
 | 1022 PineApple 1000
 | 出力
 | 1450

- データ数
 - 50*5+100*2+1000*1