

受験番号 _____ 氏名 _____

問題 1

以下は C 言語のプログラム片である (解答に関係しない include 文や以下の文を囲む関数の定義文などは省略してある)。例題を参考にして、指定された出力が得られるように各設問の空欄「A」～「C」を埋めよ。なお、解答はいずれも-1 から 5 までの整数のどれかである。

(例題) 出力：1, 4, 9, 16, <pre>int i, j; for (j = 「A」; j < 「B」; j++) { i = j * j; printf("%d", i); }</pre>	(設問 1) 出力：2, 5, 8, 11, <pre>int i, j; for (j = 「A」; j < 「B」; j++) { i = 3 * j + 「C」; printf("%d", i); }</pre>
(設問 2) 出力：1, -2, 3, -4, 5, -6, <pre>int j; for (j = 1; j < 7; j++) { printf("%d", j%「A」 ? j*「B」 : j*「C」); }</pre>	(設問 3) 出力：1, 2, 4, 8, 16, <pre>int j; for (j = 「A」; j < 5; j++) { printf("%d", 「B」 << j); }</pre>
(設問 4) 出力：0112223333 <pre>int i, j; for (i = 「A」; i < 「B」; i++) { for (j = 「C」; j <= i; j++) { printf("%d", i); } }</pre>	(設問 5) 出力：1, 2, 0, 1, 2, 0, 1, 2, 0, <pre>int i[4] = {3, 0, 1, 2}; int j[4] = {1, 2, 0, 3}; int k; for (k = 0; k < 9; k++) { printf("%d", i[j[k % 「A」] + 「B」]); }</pre>

解答欄

例題「A」	1	例題「B」	5
設問 1「A」 設問 1「C」		設問 1「B」	
設問 2「A」 設問 2「C」		設問 2「B」	
設問 3「A」		設問 3「B」	
設問 4「A」 設問 4「C」		設問 4「B」	
設問 5「A」		設問 5「B」	

受験番号 _____

氏名 _____

問題 2

以下のプログラム片がコンパイル時に構文エラーや型に関する警告が表示されないように解答欄の選択肢を選び記号を丸で囲め。

(例題) <pre>void f(「A」); /* 関数 f のプロトタイプ宣言 */ int x; ... f(x);</pre>	設問 1 <pre>「A」 f(「B」); /* 関数 f のプロトタイプ宣言 */ 「C」 x; ... **x = 1; *x = f(*f(**x));</pre>
設問 2 <pre>int f(「A」); /* 関数 f のプロトタイプ宣言 */ 「B」 x; 「C」 y[10]; ... *x = f(*x); *(y + f((「D」)y[0])) = f((「E」)y);</pre>	設問 3 <pre>struct S { 「A」 next; int x; }; struct S s1; 「B」 s2; 「C」 f(「D」); /* 関数 f のプロトタイプ宣言 */ ... s2.next = f(f(&s1));</pre>

解答欄

例題「A」:	ア. x	イ. int	ウ. int *	エ. * int	
設問 1「A」:	ア. *	イ. void	ウ. int *	エ. int	オ. int **
設問 1「B」:	ア. *	イ. void	ウ. int *	エ. int	オ. int **
設問 1「C」:	ア. *	イ. void	ウ. int *	エ. int	オ. int **
設問 2「A」:	ア. ** int	イ. * int	ウ. int	エ. int *	オ. &int
設問 2「B」:	ア. ** int	イ. * int	ウ. int	エ. int *	オ. &int
設問 2「C」:	ア. ** int	イ. * int	ウ. int	エ. int *	オ. &int
設問 2「D」:	ア. ** int	イ. * int	ウ. int	エ. int *	オ. &int
設問 2「E」:	ア. ** int	イ. * int	ウ. int	エ. int *	オ. &int
設問 3「A」:	ア. struct *	イ. struct S	ウ. struct S*	エ. struct S**	オ. struct *S
設問 3「B」:	ア. struct *	イ. struct S	ウ. struct S*	エ. struct S**	オ. struct *S
設問 3「C」:	ア. struct *	イ. struct S	ウ. struct S*	エ. struct S**	オ. struct *S
設問 3「D」:	ア. struct *	イ. struct S	ウ. struct S*	エ. struct S**	オ. struct *S

受験番号 _____ 氏名 _____

問題 3

設問 1

n 個の整数が小さい順に配列 $a[0], \dots, a[n-1]$ に格納され、同様に m 個の整数が小さい順に配列 $b[0], \dots, b[m-1]$ に格納されているとき、配列 a と配列 b のすべての整数をまとめて小さい順に並べるプログラムを作成したい。下記の関数 `merge` は、配列 a と b を連結した配列 c および要素数 n, m をそれぞれ引数として与えると、配列 c の要素を小さい順に並べ替えるものである。以下の空欄ア～ウを埋めてプログラムを完成させよ。

```
#include <stdio.h>
int merge( int c[], int n, int m );

int main(void)
{
    int a[8] = { 2, 4, 7, 10, 12, 14, 60, 79 };
    int b[10] = { 3, 5, 11, 13, 15, 19, 30, 90, 92, 95 };
    int c[100];
    int n=8, m=10, i, j, k=0;

    for( i=0; i<n; i++)
        c[k++] = a[i];
    for( j=0; j<m; j++)
        c[k++] = b[j];

    merge(c, n, m);

    return 0;
}

int merge( int c[], int n, int m )
{
    int i=0, j=0, k=0, tmp[100];

    while( i < n && j < m ){
        ア
    }
    while( i < n ){
        イ
    }
    while( j < m ){
        ウ
    }
    for( k=0; k<n+m; k++)
        c[k]=tmp[k];

    return 0;
}
```

解 答 欄:

ア	
イ	
ウ	

受験番号 _____

氏名 _____

設問 2

設問 1 で作成した関数 `merge` を利用して、任意の整数型配列の要素を小さい順に並べ替えるプログラムを作成したい。次のアルゴリズムで処理をおこなう関数 `mergesort` を、以下の空欄アを埋めて完成させよ。

- 1) 引数として配列 `d` とその要素数を受け取る。
- 2) 配列 `d` を左右に 2 分割する。
- 3) 左側の配列を引数として関数 `mergesort` を再帰的に呼び出す。
- 4) 同様に、右側の配列を引数として関数 `mergesort` を再帰的に呼び出す。
- 5) 左右の配列の要素が、それぞれ小さい順に整列されて戻ってくるので、関数 `merge` でそれらの要素をまとめて小さい順に並べ替え、配列 `d` に格納してリターンする。

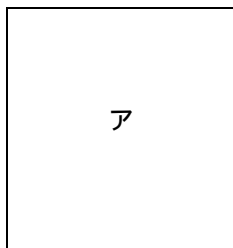
```
#include <stdio.h>
int mergesort( int d[], int len );
int merge( int c[], int n, int m );
```

```
int main(void)
{
    int d[11] = { 10, 2, 6, 0, 7, 8, 5, 4, 3, 9, 1 };
    int n=11;

    mergesort(d, n);

    return 0;
}
```

```
int mergesort( int d[], int len )
{
```



```
    return 0;
}

int merge( int c[], int n, int m )
{
    ( 設問 1 で求めた関数 )

}
```

解 答 欄:

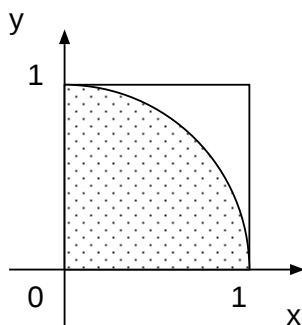
ア	
---	--

受験番号 _____ 氏名 _____

問題 4

xy 平面上の正方形領域 ($0 \leq x < 1$, $0 \leq y < 1$) 内にランダムに点をとると, その点が $(0, 0)$ を中心とした半径 1 の円内にはいる確率は, 正方形と $1/4$ 円の面積の比 $\pi/4$ に等しい. これを利用すると乱数を用いて円周率 π の近似値を求めることができる. 以下のアルゴリズムにしたがって, 円周率 π の近似値を求めて表示するプログラムを作成せよ. ただし, $0 \leq r < 1$ の範囲の一様乱数 r を戻り値として返す実数型の関数 `rrand()` が利用できるとしてよい.

- 1) 関数 `rrand()` を用いて 2 つの乱数を発生させ x, y とする.
- 2) 点 (x, y) が $(0, 0)$ を中心とした半径 1 の円内に存在するか否かを判定する.
- 3) 上記 1) ~ 2) を 10000 回反復し, 点が円内に存在する割合から円周率 π を推定する.



(解 答 欄)