

受験番号 \_\_\_\_\_ 氏名 \_\_\_\_\_

### 問題 1

- 1) 以下の文の空欄にあてはまる最も適切な語句を，語群から選択して解答欄に記入せよ．ただし，同じ番号の空欄には同じ語句が入り，異なる番号の空欄にはそれぞれ異なる語句が入るものとする．

コンピュータを動作させるには適切な (1) を与える必要がある．(1) を作成する一連の作業を (2) という．そのためには，まず (3) と呼ばれるソフトウェアを利用して (4) を入力する．次に，(5) と呼ばれるソフトウェアを利用して，(4) をコンピュータが理解可能な機械語に変換する．こうして完成した実行プログラムをコンピュータに与え，目的とする処理を実行させる．

C 言語のプログラムは，通常多くの関数から構成されている．関数とは，呼び出し側から渡された (6) に対して何らかの処理を行って，ひとつの (7) を返す機能である．ただし，(6) や (7) を持たない関数も存在する．C 言語は，関数を呼び出すときに，標準では値による呼び出しをおこなう．

#### 語群

コンパイラ，デバッガ，アドレス，プログラム，ライブラリ，アルゴリズム，コメント，プログラミング，マクロ，ヘッダ，エディタ，ソースプログラム，戻り値，初期化，式，引数，定数，実数，書式，流れ図，演算子

(1)		(2)	
(3)		(4)	
(5)		(6)	
(7)			

- 2) 「値による呼び出し」とは，(6) をどのように関数に渡す方法であるか説明せよ．

- 3) 「値による呼び出し」の利点を述べよ．

受験番号 \_\_\_\_\_

氏名 \_\_\_\_\_

## 問題 2

1 より大きな整数が入力されたときに、その整数の素因数分解を行うプログラムを作成したい。たとえば、60 が入力されたならば、 $60 = 2 \times 2 \times 3 \times 5$  と素数の積に分解できるので、これを「 2 2 3 5 」と出力するプログラムを作成する。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 main(void)
{
    int n, i;

    scanf("%d", &n);

    i = 2;
    while(  ){
        if( n%i == 0 ){
            printf("%d ",i);
            
        }
        else{
            
        }
    }
    printf("\n");
    return 0;
}
```

解 答 欄 :

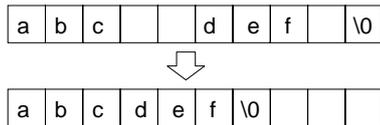
ア	
イ	
ウ	

受験番号 \_\_\_\_\_ 氏名 \_\_\_\_\_

### 問題 3

#### 設問 1

任意の文字列が与えられたとき, その中から空白を取り除き, 図のように空白以外の文字を左に詰めた文字列に変換する関数 `delspace()` を, 以下の空欄を埋めて完成させよ.



```
#include <stdio.h>

int delspace( char str[] );

int main(void)
{
    char moji[100] = "abc def ";

    printf("%s\n", moji);

    delspace(moji);

    printf("%s\n", moji);

    return 0;
}

int delspace( char str[] )
{
    

ア



    return 0;
}
```

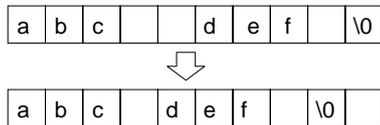
解 答 欄:

ア	
---	--

受験番号 \_\_\_\_\_ 氏名 \_\_\_\_\_

設問 2 (問題 3)

任意の文字列が与えられたとき, 2 つ以上の空白が連続していれば, その部分を図のように空白 1 つに変換する関数 `shrinkspace()` を, 以下のコードに続けて完成させよ.



解答欄:

```
int shrinkspace( char str[] )  
{
```

受験番号 \_\_\_\_\_ 氏名 \_\_\_\_\_

## 問題 4

### 設問 1

$x$  と  $y$  の 2 つの整数要素を持つ 2 次元ベクトルを処理するプログラムを、以下のような構造体を使って作成することを考える。

```
struct vector {  
    int x;  
    int y;  
};
```

さて、このプログラムでは 2 つのベクトル間に次のような大小関係を定義してソートを行うこととする。

- 1)  $x^2 + y^2$  の値の大きな方のベクトルを「大きい」とする
- 2) 1) が等しい場合には、 $x$  の値の大きな方のベクトルを「大きい」とする
- 3) 1) も 2) も等しい場合には、 $y$  の値の大きな方のベクトルを「大きい」とする
- 4) 1)~3) でも大小が決まらなかった場合には、2 つのベクトルは「等しい」とする

この大小関係を判定するための関数を作成したい。引数として 2 つのベクトルへのポインタ  $a$ ,  $b$  をとり、 $a$  が大きい場合には 1 を、 $b$  が大きい場合には  $-1$  を、 $a$  と  $b$  が等しい場合には 0 を返す関数 `vectorcmp()` を、以下の記述に続けて完成させよ。

```
int vectorcmp(struct vector *a, struct vector *b)  
{
```

受験番号 \_\_\_\_\_ 氏名 \_\_\_\_\_

## 設問 2 (問題 4)

設問 1 で定義した大小関係に従って昇順に整列した 100 個のベクトルのデータが外部宣言された配列 `table` に格納されている。検索対象となるベクトルが 1 つ与えられたとき、これが `table` の中にあるかどうかを二分探索で調べたい。二分探索とは、配列の中央のデータと検索対象のデータを比較し、目的のデータが中央のデータを境にしてどちらの側にあるのかを判断し、これを繰り返すことによって、検索対象データが存在し得る範囲を狭めていく探索手法である。引数として探索目的のベクトルへのポインタ `target` をとり、見つければ 1 を見つからなければ 0 を返す関数 `search()` を以下の記述に続けて完成させよ。なお、必要があれば設問 1 で作成した関数 `vectorcmp()` を使用してよい。

```
struct vector table[100];  
  
...  
  
int search(struct vector *target)  
{
```