

令和 7 年度
山梨大学 大学院医工農学総合教育部 修士課程 工学専攻

入 学 試 験 問 題

No. 1

コース等	コンピュータ理工学 コース	試 験 分 野	専 門 科 目
------	------------------	---------	---------

試験時間は 1 時間 30 分です。試験監督から指示があるまで、この表紙をめくってはいけません。
次ページ以降に問題が、右上に番号付けされた用紙に分けて出題されています。配点は 140 点です。
解答にあたっては、解答用紙の表紙の指示に従いなさい。

解答開始の合図の後、各分野の問題について下表中に示す No. の用紙が綴じ込まれていることを確認しなさい。用紙に乱丁・落丁がある場合には、手を挙げて試験監督に知らせなさい。

	分野名	問題用紙の ページ番号
必須	アルゴリズムとデータ構造, 並びにプログラミング	No. 2~9

解答は、原則、1 問につき 1 枚を使用する。もしも解答用紙のスペースが不足の場合には、手を挙げて試験監督に知らせること。

すべての解答用紙について、受験番号欄に受験番号を記入の上で試験終了後に提出しなさい。本用紙を含むすべての問題用紙についても、回収します。

令和 7 年度
山梨大学 大学院医工農学総合教育部 修士課程 工学専攻

入 学 試 験 問 題

No 2

コ ー ス 等	コンピュータ理工学 コース	試 験 科 目	アルゴリズムとデータ構造, 並びにプログラミング
---------	------------------	---------	-----------------------------

問 1 No3 ページの C++言語で書かれたソースコード<sum_range.cpp>は、下記の<実行結果 1>から<実行結果 5>のとおり、標準入力から入力された変数 a, b について、a から b までの整数の和を標準出力へ出力するコードの一部である。下記の実行結果のとおりとなるよう、再帰関数 `int sum_range( )` における空欄 A と空欄 B に当てはまる適切なコードを記述しなさい。ただし、この関数の引数リスト以外の変数を宣言してはならないこととする。なお、標準入力からの a, b への入力においては、必ず $a < b$ となる整数値が入力されることを前提としてかまわない。

※実行結果中の下線部は、標準入力からの入力を示す。

<実行結果 1>

a=1

b=10

The sum of integers from 1 to 10 is 55.

<実行結果 2>

a=4

b=9

The sum of integers from 4 to 9 is 39.

<実行結果 3>

a=11

b=12

The sum of integers from 11 to 12 is 23.

<実行結果 4>

a=-6

b=-2

The sum of integers from -6 to -2 is -20.

<実行結果 5>

a=-99

b=99

The sum of integers from -99 to 99 is 0.

令和 7 年度
山梨大学 大学院医工農学総合教育部 修士課程 工学専攻

入 学 試 験 問 題

No 3

コ ー ス 等	コンピュータ理工学 コース	試 験 科 目	アルゴリズムとデータ構造, 並びにプログラミング
---------	------------------	---------	-----------------------------

<sum_range.cpp>

```
#include <iostream>
using namespace std;

int sum_range(  )
{
    
}

int main()
{
    int a=0, b=0;

    cout << "a="; cin >> a;
    cout << "b="; cin >> b;

    cout << "The sum of integers from " << a << " to " << b
        << " is " << sum_range(a, b) << ".\n";

    return 0;
}
```

令和 7 年 度
山梨大学 大学院医工農学総合教育部 修士課程 工学専攻

入 学 試 験 問 題

No 4

コ ー ス 等	コンピュータ理工学 コース	試 験 科 目	アルゴリズムとデータ構造, 並びにプログラミング
---------	------------------	---------	-----------------------------

問 2 No 4 から No 7 ページまでの C++言語で書かれたソースコード<ID_validator.cpp>は、とある大学の学籍番号の文字列を検証するコードと、その動作確認結果を標準出力へ出力するコードである。このソースコードについて、以下の設問に答えなさい。なお、コード中の関数 `stoi( )` は、引数の文字列を数値として読み取り、`int` 型の値に変換して戻り値として返す関数である。

- (1) No 4 ページの//C の行の `virtual bool validate( )` のような仮想関数を使用する理由について、このソースコードを例に 150 字程度で説明しなさい。
- (2) コンパイル後、実行しているときの標準出力への出力をすべて示しなさい。
- (3) No 5 ページの//D, //E, //F の行および No 6 ページの//G, //H の行の「`unique_ptr`」は、動的にオブジェクトを管理する「スマートポインタ」である。このようなスマートポインタを使用する場合に対比して、スマートポインタを使わずに動的なメモリを確保する「`new` 演算子」および解放する「`delete` 演算子」を使用した場合の問題点をひとつ挙げ、それと比べたスマートポインタの利点とあわせて 100 字程度で説明しなさい。

<ID_validator.cpp>

```
#include <string>
#include <memory>
#include <iostream>
using namespace std;

class studentID_validator
{
public:
    virtual bool validate(string studentID) = 0;    //C
    virtual ~studentID_validator() { }
};

class length_validator : public studentID_validator
{
    const unsigned int length = 8;
public:
    length_validator() { }

    bool validate(string studentID) override {
        return studentID.length() == length;
    }
};
```

//次ページへつづく

令和 7 年度
山梨大学 大学院医工農学総合教育部 修士課程 工学専攻

入 学 試 験 問 題

No 5

コ ー ス 等	コンピュータ理工学 コース	試 験 科 目	アルゴリズムとデータ構造, 並びにプログラミング
---------	------------------	---------	-----------------------------

//前ページからのつづき

```
class studentID_validator_decorator : public studentID_validator
{
    unique_ptr<studentID_validator> inner;                //D
public:
    studentID_validator_decorator(unique_ptr<studentID_validator>
validator) :
        inner(move(validator)) { }

    bool validate(string studentID) override {
        return inner->validate(studentID);
    }
};

class t_validator : public studentID_validator_decorator
{
public:
    t_validator(unique_ptr<studentID_validator> validator) :    //E
        studentID_validator_decorator(move(validator)) { }

    bool validate(string studentID) override {
        if (!studentID_validator_decorator::validate(studentID))
            return false;
        return studentID[0] == 't';
    }
};

class year_validator : public studentID_validator_decorator
{
public:
    year_validator(unique_ptr<studentID_validator> validator) :    //F
        studentID_validator_decorator(move(validator)) { }

    bool validate(string studentID) override {
        if (!studentID_validator_decorator::validate(studentID))
            return false;
        return stoi(studentID.substr(1, 2)) > 20
            && stoi(studentID.substr(1, 2)) < 25;
    }
};
```

//次ページへつづく

令和 7 年度
山梨大学 大学院医工農学総合教育部 修士課程 工学専攻

入 学 試 験 問 題

No 6

コ ー ス 等	コンピュータ理工学 コース	試 験 科 目	アルゴリズムとデータ構造, 並びにプログラミング
---------	------------------	---------	-----------------------------

//前ページからのつづき

```
class cs_validator : public studentID_validator_decorator
{
public:
    cs_validator(unique_ptr<studentID_validator> validator) :    //G
        studentID_validator_decorator(move(validator)) { }

    bool validate(string studentID) override {
        if (!studentID_validator_decorator::validate(studentID))
            return false;
        return studentID[3] == 'c' && studentID[4] == 's';
    }
};

class num_validator : public studentID_validator_decorator
{
public:
    num_validator(unique_ptr<studentID_validator> validator) :    //H
        studentID_validator_decorator(move(validator)) { }

    bool validate(string studentID) override {
        if (!studentID_validator_decorator::validate(studentID))
            return false;
        return stoi(studentID.substr(5, 3)) > 0
            && stoi(studentID.substr(5, 3)) < 100;
    }
};
```

//次ページへつづく

令和 7 年度
山梨大学 大学院医工農学総合教育部 修士課程 工学専攻

入 学 試 験 問 題

No 7

コ ー ス 等	コンピュータ理工学 コース	試 験 科 目	アルゴリズムとデータ構造, 並びにプログラミング
---------	------------------	---------	-----------------------------

//前ページからのつづき

```
int main()
{
    auto validator1 = make_unique<length_validator>();
    cout << "(1)";
    if (validator1->validate("abc12345")) cout << "valid\n";
    else cout << "invalid\n";

    auto validator2 = make_unique<t_validator>(
        make_unique<length_validator>());
    cout << "(2)";
    if (validator2->validate("tab12345")) cout << "valid\n";
    else cout << "invalid\n";

    auto validator3 = make_unique<year_validator>(
        make_unique<t_validator>(make_unique<length_validator>()));
    cout << "(3)";
    if (validator3->validate("t19cs099")) cout << "valid\n";
    else cout << "invalid\n";

    auto validator4 = make_unique<num_validator>(
        make_unique<cs_validator>(make_unique<year_validator>(
            make_unique<t_validator>(make_unique<length_validator>()))));
    cout << "(4)";
    if (validator4->validate("t24cs099")) cout << "valid\n";
    else cout << "invalid\n";

    return 0;
}
```

令和 7 年度
山梨大学 大学院医工農学総合教育部 修士課程 工学専攻

入 学 試 験 問 題

No. 8

コ ー ス 等	コンピュータ理工学 コース	試 験 科 目	アルゴリズムとデータ構造, 並びにプログラミング
---------	------------------	---------	-----------------------------

問 3 以下の問いに答えなさい.

$(x+y)^n$ を展開すると以下のとおりとなる.

$$(x+y)^0 = 1 \quad (= 1 \cdot x^0 \cdot y^0)$$

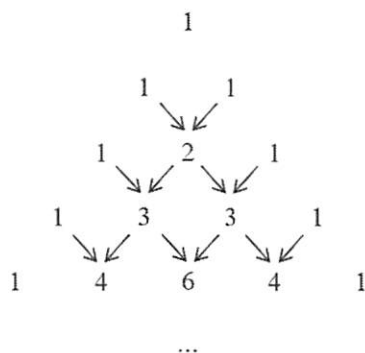
$$(x+y)^1 = x + y$$

$$(x+y)^2 = x^2 + 2xy + y^2$$

$$(x+y)^3 = x^3 + 3x^2y + 3xy^2 + y^3$$

...

このときの各項の係数はそれぞれ, 下図のようなパスカルの三角形で表すことができる.



- (1) $(x+y)^n$ における $x^i y^{n-i}$ の係数 a_i^n を, パスカルの三角形を参照して漸化式の形で示しなさい.
- (2) $(x+y)^n$ の各項の係数がパスカルの三角形で表すことができることを, 数学的帰納法で証明しなさい.
- (3) 木は連結で閉路を持たないグラフであるが, 木の頂点の数 $|V|$, 辺の数 $|E|$ に対して以下の式が成り立つ.

$$|E| = |V| - 1$$

このことを数学的帰納法で証明しなさい.

令和 7 年度
山梨大学 大学院医工農学総合教育部 修士課程 工学専攻

入 学 試 験 問 題

No. 9

コ ー ス 等	コンピュータ理工学 コース	試 験 科 目	アルゴリズムとデータ構造, 並びにプログラミング
---------	------------------	---------	-----------------------------

問4 以下の0-1ナップザック問題を動的計画法で解くことを考える.

目的関数: $c_1x_1 + c_2x_2 + \dots + c_nx_n$ (最大化)

制約条件: $a_1x_1 + a_2x_2 + \dots + a_nx_n \leq b$

ただし, $x_i \in \{0, 1\}$ とし, また b と各 a_i は非負整数とする.

以下のような表を, 動的計画法を用いて計算して最適解を求める. ここで変数 $z_k(p)$ は, 容量 p のナップザックに, 品物 1, ..., k から幾つかのものを選択して詰め込んだ場合の目的関数の最大値を表すものとする. また変数 $type_k(p)$ は, その最適解を得るために品物 k が必要か否かを示す真偽値変数である.

	k=1		k=2		k=3		k=4	
p	$z_1(p)$	$type_1(p)$	$z_2(p)$	$type_2(p)$	$z_3(p)$	$type_3(p)$	$z_4(p)$	$type_4(p)$
0								
1								
2								
3								
4								
5								
6								

(1) このとき $z_k(p)$ は以下の漸化式を用いて計算できる. 式中の空欄A, B, C を答えなさい.

$$1. \quad z_1(p) = \begin{cases} -\infty & \text{if } p < 0 \\ \boxed{A} & \text{if } 0 \leq p < a_1 \\ c_1 & \text{otherwise} \end{cases}$$

$$2. \quad z_k(p) = \max(z_{k-1}(\boxed{B}) + c_k, z_{k-1}(\boxed{C})) \quad \text{if } k > 1$$

(2) 下記のナップザック問題を動的計画法を用いて解き, 最適解を答えなさい. 計算過程を示すために, 解答用紙に示した表も埋めなさい.

目的関数: $2x_1 + 1x_2 + 3x_3 + 3x_4$ (最大化)

制約条件: $4x_1 + 3x_2 + 2x_3 + x_4 \leq 6$

- (3) 0-1 ナップザック問題の品物(変数)の数を n , ナップザックの容量を b とするとき, 動的計画法を適用する表の大きさを概数(byte数)で答えなさい. 逆に, メインメモリが2 GiB 利用できる場合に, 動的計画法ではどの程度の規模のナップザック問題が解けるかを答えなさい. 同時に, CPUのクロック周波数を2 GHzとした場合に必要となるおおよその計算時間を考察し, 説明しなさい. ただし, 考察する上での数量的な仮定を明記して説明すること.
- (4) 動的計画法の特徴を, 0-1ナップザック問題へ分枝限定法を適用した場合と比較して考察し, それぞれの長所と短所を含めて説明しなさい.

【後期募集】

令和7年度
山梨大学 大学院医工農学総合教育部

修士課程（工学専攻） 後期募集

受験番号

入学試験解答用紙

コース等	コンピュータ理工学コース		
試験科目	アルゴリズムとデータ構造, 並びにプログラミング	採点	

問1 解答 （注意：各問について各1枚の解答用紙を使用すること。）

※ 裏面も使用するときは、☒を付して下さい。

☐ 裏面あり

【後期募集】

令和7年度
山梨大学 大学院医工農学総合教育部

修士課程（工学専攻） 後期募集

受験番号

入学試験解答用紙

コース等	コンピュータ理工学コース		
試験科目	アルゴリズムとデータ構造, 並びにプログラミング	採点	

問2 解答 （注意：各問について各1枚の解答用紙を使用すること。）

※ 裏面も使用するときは、☒を付して下さい。

☐ 裏面あり

【後期募集】

令和7年度
山梨大学 大学院医工農学総合教育部

修士課程（工学専攻） 後期募集

受験番号

入学試験解答用紙

コース等	コンピュータ理工学コース		
試験科目	アルゴリズムとデータ構造, 並びにプログラミング	採点	

問3 解答 （注意：各問について各1枚の解答用紙を使用すること。）

※ 裏面も使用するときは、☒を付して下さい。

☐ 裏面あり

【 後 期 募 集 】

令和7年度
山梨大学 大学院医工農学総合教育部

修士課程（工学専攻） 後 期 募 集

受験番号

入 学 試 験 解 答 用 紙

コース等	コンピュータ理工学コース		
試験科目	アルゴリズムとデータ構造, 並びにプログラミング	採 点	

問 4 解 答 （注意：各問について各1枚の解答用紙を使用すること。）

(1)

(2)

	$k=1$		$k=2$		$k=3$		$k=4$	
p	$z_1(p)$	$type_1(p)$	$z_2(p)$	$type_2(p)$	$z_3(p)$	$type_3(p)$	$z_4(p)$	$type_4(p)$
0								
1								
2								
3								
4								
5								
6								

(3)

(4)

※ 裏面も使用するときは、☒を付して下さい。

☐ 裏面あり