

# 山梨大学工学部コンピュータ理工学科 令和4年度3年次編入学試験説明資料

コンピュータ理工学科

3年次編入学生の選抜試験では、提出された成績証明書の内容ならびに本学で実施しました試験の結果を総合して判定し、合格者を決定しました。

令和3年6月12日に実施しました3年次編入学試験の概要は次の通りです。

## 1. 筆記試験

プログラミング、計算機アーキテクチャ、情報数学から2科目を選択して解答してもらいました。解答時間は80分です。試験問題は別紙の通りです。

## 2. 口述試験

コンピュータ理工学に関する専門分野の基礎的事項、意欲、コミュニケーション力、思考力を試問しました。試験時間は10分です。

令和4年度 山梨大学工学部

3 年 次 編 入 学 筆 記 試 験 問 題 (表紙)  
(一般選抜)

コンピュータ理工学科

プログラミング, 計算機アーキテクチャ, 情報数学から2科目を選択して解答用紙に解答してください。解答にあたっては, 解答用紙の表紙の指示に従ってください。

解答開始の合図の後, 各科目の問題用紙が下表中に示すページ No. のとおりに配布されていることを確認してください。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てください。

科目名	問題用紙の枚数とページ No.
プログラミング	7 枚 (No. 1/7, No. 2/7, No. 3/7, No. 4/7, No. 5/7, No. 6/7, No. 7/7)
計算機アーキテクチャ	2 枚 (No. 1/2, No. 2/2)
情報数学	1 枚 (No. 1/1)

3 年 次 編 入 学 筆 記 試 験 問 題

No 1 / 7

学 科	コンピュータ理工学科	試 験 科 目	プログラミング
-----	------------	---------	---------

問 1 (配点:60 点)

仮引数として受け取った自然数  $N$  が素数 (true) か否 (false) かを戻り値として返す関数を作成する課題において、学生 A さんと学生 B さんは、それぞれ以下のコードを記述した (<C 言語>は、<C++言語>のコードに「#include <stdbool.h>」を加筆したコードである.)。これらのコードについて、以下の設問に答えなさい。

<C++言語>

```

/*学生 A さん*/
bool isPrimeA(const int N)
{
    for (int i = 2; i*i <= N; ++i) {
        if (N % i == 0) return (ア);
    }
    return (イ);
}
    
```

```

/*学生 B さん*/
bool isPrimeB(const int N)
{
    for (int i = 2; i < N; ++i) {
        if (N % i == 0) return (ウ);
    }
    return (エ);
}
    
```

<C 言語>

```

/*学生 A さん*/
#include <stdbool.h>

bool isPrimeA(const int N)
{
    for (int i = 2; i*i <= N; ++i) {
        if (N % i == 0) return (ア);
    }
    return (イ);
}
    
```

```

/*学生 B さん*/
#include <stdbool.h>

bool isPrimeB(const int N)
{
    for (int i = 2; i < N; ++i) {
        if (N % i == 0) return (ウ);
    }
    return (エ);
}
    
```

- (a) 空欄(ア)～(エ)には、題意を満たすために“true”または“false”が記述されている。この空欄(ア)～(エ)のそれぞれについて、正しく当てはまるコードを示しなさい。
- (b) 学生 A さんと学生 B さんの関数のアルゴリズムについて、それぞれの最悪の時間計算量をオーダー記法で示し、どちらの方が効率のよいアルゴリズムなのかを答えなさい。

3 年 次 編 入 学 筆 記 試 験 問 題

No 2 / 7

学 科	コンピュータ理工学科	試 験 科 目	プログラミング
-----	------------	---------	---------

問 2 (配点: 70 点)

ページ No 3/7 (同内容の C 言語版はページ No 4/7) に示した<ソースコード>は, 整数の 2 進数表現とビット演算を用いて以下の囲みに示した<部分和问题>を解くコードである.

<部分和问题>

$N$  個の正の整数  $a_0, a_1, \dots, a_{N-1}$  と正の整数  $W$  が与えられたとき,  $a_0, a_1, \dots, a_{N-1}$  の中からいくつかを選んで, その和を  $W$  とできる (Yes) か否 (No) かを出力する.

たとえば, このコードの実行結果として, `main( )` 関数の `const int W` が 19 の場合 “Yes” と出力され, 20 の場合 “No” と出力される. このソースコードについて, 以下の設問に答えなさい.

- 関数 `bool isSubsetSum( )` 内の `//(A)` の行の `bit & (1 << i)` は, この部分和问题を解くためのどのような処理をおこなっているか. この中の `&` および `<<` の機能の説明を含めて, 150 字程度で説明しなさい.
- このソースコードにおける最悪の時間計算量をオーダー記法で示し, その理由を 50 字程度で示しなさい.
- このソースコードのアルゴリズムの時間計算量よりも効率よく部分和问题を解くアルゴリズムとして, 時間計算量が  $O(NW)$  となるアルゴリズムがある. この時間計算量  $O(NW)$  となるアルゴリズムとして, どのようなアルゴリズムが考えられるか. 200 字程度で説明しなさい. 必要であれば, この説明文に加えて図を用いてかまわない.

## 3 年 次 編 入 学 筆 記 試 験 問 題

No 3/7

学 科	コンピュータ理工学科	試 験 科 目	プログラミング
-----	------------	---------	---------

&lt;ソースコード(C++言語)&gt;

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

bool isSubsetSum(const int a[], const int N, const int W)
{
    for (int bit = 0; bit < (1 << N); ++bit)
    {
        int sum = 0;
        for (int i = 0; i < N; ++i) {
            if (bit & (1 << i)) { // (A)
                sum += a[i];
            }
        }
        if (sum == W) return true;
    }

    return false;
}

int main()
{
    const int N = 4, W = 19;
    const int a[] = { 3, 5, 7, 9 };

    if (isSubsetSum(a, N, W)) cout << "Yes\n";
    else cout << "No\n";

    return 0;
}
```

## 3 年 次 編 入 学 筆 記 試 験 問 題

No 4 / 7

学 科	コンピュータ理工学科	試 験 科 目	プログラミング
-----	------------	---------	---------

&lt;ソースコード(C言語)&gt;

```
#include <stdio.h>
#include <stdbool.h>

bool isSubsetSum(const int a[], const int N, const int W)
{
    for (int bit = 0; bit < (1 << N); ++bit)
    {
        int sum = 0;
        for (int i = 0; i < N; ++i) {
            if (bit & (1 << i)) { // (A)
                sum += a[i];
            }
        }
        if (sum == W) return true;
    }
    return false;
}

int main()
{
    const int N = 4, W = 19;
    const int a[] = { 3, 5, 7, 9 };

    if (isSubsetSum(a, N, W)) printf("Yes\n");
    else printf("No\n");

    return 0;
}
```

3 年 次 編 入 学 筆 記 試 験 問 題

No 5 / 7

学 科	コンピュータ理工学科	試 験 科 目	プログラミング
-----	------------	---------	---------

問3(配点:70点)

ページ No 6/7(同内容の C 言語版はページ No 7/7)に示した<ソースコード>は, 都県名の一覧を処理するソースコードである. このソースコードについて, 以下の設問に答えなさい.

- (a) このソースコードをコンパイルして実行したときの, 標準出力への出力をすべて書き出さなさい.
- (b) `main( )`関数内の「`//(A)`」の行で呼び出される `func1( )`について, このときに渡される実引数に基づく実行の様子を説明しなさい. 必要であれば, 図を用いてかまわない.
- (c) このソースコードで都県名の一覧に用いられているデータ構造の代わりに配列の使用を考える. 設問(b)の処理においては, 時間計算量の観点から, 配列の方が効率が悪い. その理由を 150 字程度で説明しなさい.

## 3 年 次 編 入 学 筆 記 試 験 問 題

No 6 / 7

学 科	コンピュータ理工学科	試 験 科 目	プログラミング
-----	------------	---------	---------

&lt;ソースコード(C++言語)&gt;

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

class Node {
public:
    Node* next;
    string pref;
    Node(string p = "") :
        next(NULL), pref(p) { }
};

void func1(Node* v, Node* p)
{
    v->next = p->next;
    p->next = v;
}

void func2(const Node *n)
{
    Node* cur = n->next;
    while (cur != n) {
        cout << cur->pref << "->";
        cur = cur->next;
    }
    cout << endl;
}

void func3(Node *n)
{
    Node* cur = n->next;
    while (cur != n) {
        Node* temp = cur->next;
        delete cur;
        cur = temp;
    }
}

//右段へ続く
```

```
//左段からのつづき

int main()
{
    Node* np;
    np = new Node();
    np->next = np;

    const vector<string> prefs
        = { "Tokyo",
            "Kanagawa",
            "Saitama",
            "Nagano" };

    for (int i = 0; i < (int)prefs.size(); ++i) {
        Node* node = new Node(prefs[i]);
        func1(node, np);
        func2(np);
    }

    Node* node = new Node("Yamanashi");
    func1(node, np->next);    //(A)
    func2(np);

    func3(np);
    delete np;

    return 0;
}
```

## 3 年次編入学筆記試験問題

No 7/7

学 科	コンピュータ理工学科	試 験 科 目	プログラミング
-----	------------	---------	---------

&lt;ソースコード(C言語)&gt;

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define CHAR_NUM 10

typedef struct _Node {
    struct _Node* next;
    char pref[CHAR_NUM];
}Node;

void func1(Node* v, Node* p)
{
    v->next = p->next;
    p->next = v;
}

void func2(const Node* n)
{
    Node* cur = n->next;
    while (cur != n) {
        printf("%s->", cur->pref);
        cur = cur->next;
    }
    printf("※n");
}

void func3(Node* n)
{
    Node* cur = n->next;
    while (cur != n) {
        Node* temp = cur->next;
        free(cur);
        cur = temp;
    }
}
```

//右段へ続く

```
//左段からのつづき

int main()
{
    Node* np;
    np = malloc(sizeof(Node));
    strcpy(np->pref, "");
    np->next = np;

    const char prefs[][CHAR_NUM]
        = { "Tokyo",
            "Kanagawa",
            "Saitama",
            "Nagano" };

    for (int i = 0; i < (int)sizeof(prefs) /
        sizeof(prefs[0]); ++i) {
        Node* node = malloc(sizeof(Node));
        strcpy(node->pref, prefs[i]);
        func1(node, np);
        func2(np);
    }

    Node* node = malloc(sizeof(Node));
    strcpy(node->pref, "Yamanashi");
    func1(node, np->next);    //(A)
    func2(np);

    func3(np);
    free(np);

    return 0;
}
```

3 年 次 編 入 学 筆 記 試 験 問 題

No 1 / 2

学 科	コンピュータ理工学科	試 験 科 目	計算機アーキテクチャ
-----	------------	---------	------------

問 1 (配点: 120 点)

4bit の 2 進数  $x = (a, b, c, d)$  があり, その 10 進数表記は

$$x = a \times 2^3 + b \times 2^2 + c \times 2^1 + d \times 2^0$$

であるとする. 論理関数  $y = f(x)$  の出力  $y$  は, 入力  $x$  を 10 進数表記したとき,  $x$  が 4 または 5 の倍数 (0 は除く) のときに 1 となり, それ以外のときに 0 となる. このとき  $f$  を実現する論理回路を次の手順で設計しなさい.

- (a) 入力変数  $a, b, c, d$  と出力変数  $y$  に関する真理値表を, 右図の真理値表を解答用紙に書き写してから作成しなさい.

$a$	$b$	$c$	$d$	$y$
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

- (b) 上記(a)の真理値表をもとに,  $f$  に対する主加法標準形 (積和標準形) の論理式を求めなさい.

- (c) 論理関数  $f$  を表す論理式の最簡形を, 以下の真理値表をもとにカルノー図を作成して求めなさい.

$cd$	00	01	11	10
$ab$				
00				
01				
11				
10				

- (d) 上記(c)で求めた論理式を計算する論理回路を描きなさい. なお論理素子は 4 入力1出力の AND 素子 , 3入力1出力の AND 素子 , 4入力1出力の OR 素子 , 1入力1出力の NOT 素子  の 4 種類だけを使用し, 図表記もこの説明文内のものを用いること. 使用される素子数は問わない. また入力端子は余らせずに全て接続すること.

3 年 次 編 入 学 筆 記 試 験 問 題

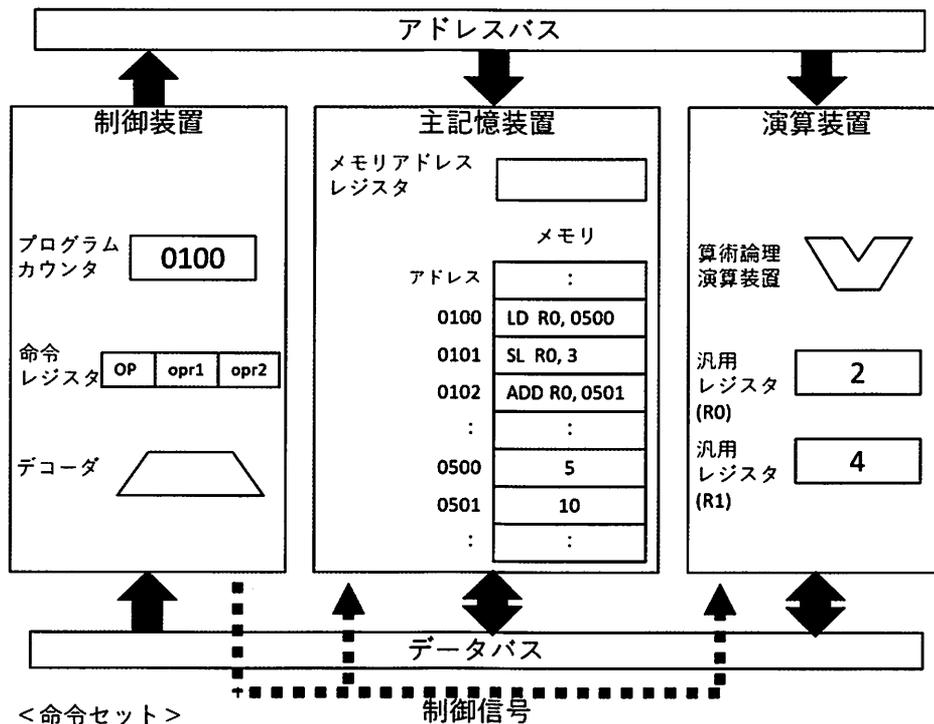
No 2 / 2

学 科	コンピュータ理工学科	試 験 科 目	計算機アーキテクチャ
-----	------------	---------	------------

問 2 (配点:80 点)

下図のノイマン型コンピュータに関して以下の設問に答えなさい。なお図中の数値は 10 進数で表記する。演算には 2 進法を用いる。プログラムカウンタの値は 1 ずつ増えることにする。レジスタ、メモリアドレス、メモリデータは全て 32 ビット固定長とする。

- (a) 以下の説明文の(ア)～(エ)を埋めなさい。  
 制御装置の(ア)に格納されているアドレスを主記憶装置の(イ)に送る。(イ)が参照しているアドレスに格納されている命令を(ウ)に取り出し、(エ)に送り解読する。(エ)は命令に応じた制御信号を演算装置に送る。演算装置は制御信号に従って命令を実行する。命令の実行完了後、(ア)に格納されているアドレスを更新する。以上の作業を繰り返す。
- (b) プログラムカウンタと各種レジスタ、メモリの内容が下図の通りである。0100 番地から 0102 番地のメモリに格納したプログラムを実行した際の汎用レジスタ (R0) の値 (10 進数) の変化を答えなさい。



< 命令セット >

命令	命令の意味
LD opr1, opr2	opr2が示すアドレスの内容をopr1が示すレジスタに転送する。
SL opr1, opr2	opr1がレジスタの値をビット列としてopr2桁だけ左にずらす。
ADD opr1, opr2	opr2が示すアドレスの内容をopr1が示すレジスタの値に加算する。

注) OP: 命令コード, opr1:オペランド1, opr2:オペランド2

3 年 次 編 入 学 筆 記 試 験 問 題

No 1 / 1

学 科	コンピュータ理工学科	試 験 科 目	情報数学
-----	------------	---------	------

問 1 (配点:100 点)

2 種の記号 A と B の発生確率がそれぞれ 0.8 と 0.2 である記憶のない情報源  $S$  を考える. このとき, 以下の設問に答えなさい. 計算には,  $\log_2 10 = 3.322$  を用いてよい. また, 答えは小数点以下第 4 位を四捨五入して表しなさい.

- (a)  $S$  のエントロピーを求めなさい.
- (b)  $S$  を 2 次に拡大してからハフマン符号化をした場合に生成される符号を示しなさい. また, 1 情報源記号当たりの平均符号長を求めなさい.
- (c) A と B の発生確率が偏っているので, ランレングス符号化を行う. ただし, A は長さ 4 までのランを考え, B は長さ 1 のランのみを考える. すなわち, 下記の 5 種類のランを考える. これをハフマン符号化し, 生成される符号を示しなさい. また, 1 情報源記号当たりの平均符号長を求めなさい.

AAAA, AAAB, AAB, AB, B

- (d) (c) の符号化は (b) の符号化に比べて, どれだけ符号化の効率 (1 情報源記号当たりの平均符号長に対するエントロピーの比) が改善されているか答えなさい.

問 2 (配点:100 点)

行列  $A = \begin{pmatrix} 2 & 1 & -1 \\ 1 & 0 & -1 \\ 1 & -1 & 0 \end{pmatrix}$  について, 以下の設問に答えなさい.

- (a)  $A$  の固有値, 固有ベクトルを求めなさい.
- (b)  $\begin{pmatrix} -1 \\ -3 \\ -1 \end{pmatrix}$  を  $A$  の固有ベクトルの 1 次結合で表しなさい.
- (c) 自然数  $n$  に対して,  $A^n \begin{pmatrix} -1 \\ -3 \\ -1 \end{pmatrix}$  を求めなさい.