

令和8年4月入学（第2回）

山口大学大学院創成科学研究科（工学系）博士前期課程入学試験

受験区分コード54 専門科目（データ構造とアルゴリズム(その1)）

以下の問に答え、解答を解答用紙（データ構造とアルゴリズム(その1 解答)）に記入しなさい。

問1（配点 50点）

枠内は、2つの文字列 s_1 と s_2 が与えられたときの編集距離（何文字操作すれば同じ単語となるか）を求める再帰関数 $D(i, j)$ の処理である。 s_1 の文字列長が l_1 、 s_2 の文字列長が l_2 の場合は、 $D(l_1, l_2)$ として、この関数が適用されるとして、以下の問に答えよ。また、文字列"abc"の"a"を1文字目、文字列長は3とみなす。

関数 $D(i, j)$ は、順に Step. 1~5.の指示を実行する。

Step 1. i が0の時は、 j を編集距離とし、Step 5へ。

Step 2. j が0の時は、 i を編集距離とし、Step 5へ。

Step 3. s_1 の i 番目の文字と s_2 の j 番目の文字が一致するなら変数 $cost$ を0とし、それ以外は $cost$ を1とする。

Step 4. $D(i-1, j-1) + cost$, $D(i, j-1) + 1$, $D(i-1, j) + 1$ を順に求め、その計算結果の中で最小値を編集距離とする。

Step 5. 求まった編集距離を戻り値として返す。

例題： $s_1 = "a"$, $s_2 = "ab"$ の文字列の編集距離を $D(1,2)$ として求めた場合、編集距離は1となる。ここでは、呼び出される関数及びその引数と、その Step 4 で計算される式及び Step 5 の戻り値(return)を以下のように順に記述し、実行過程を示している。ただし、関数が同一の引数で呼び出される場合は、1回だけ記述している。

```

D(1,2) D(0,1)+1, D(1,1)+1, D(0,2)+1 return 1
D(0,1) return 1
D(1,1) D(0,0)+0, D(1,0)+1, D(0,1)+1 return 0
D(0,0) return 0
D(1,0) return 1
D(0,2) return 2

```

- 関数 $D(i, j)$ を適用し、 $s_1 = "abc"$, $s_2 = "a"$ の文字列の編集距離を求めた場合、この計算過程で、再帰的に呼び出される全ての関数とその Step 4 の計算と戻り値を答えなさい。ここでは、例題に記載の方法で、実行過程を記述せよ。
- 関数 $D(i, j)$ を適用し、 $s_1 = "dog"$, $s_2 = "bag"$ の文字列の編集距離を求めた場合、この計算過程で、再帰的に呼び出される全ての関数とその Step 4 の計算と戻り値を全て答えなさい。ここでは、例題に記載の方法で、実行過程を記述せよ。
- 実行過程において、ある引数で呼び出された関数の計算結果を保存しておき、次に、同一の引数で関数が呼び出された場合は、保存した計算結果を再利用することで、関数の処理を省略することができる。この再利用を実現する以下の Step X を関数に加える場合、どこに挿入すれば、最も処理を減らすことができるか、選択肢 A)~D)の中から選び答えなさい。

"Step X. 一度、計算した引数の組合せの場合、保存した計算結果を編集距離として、Step 5へ。"

- Step 1 の前
- Step 2 の前
- Step 3 の前
- Step 4 の前

以下の問に答え、解答を解答用紙（データ構造とアルゴリズム(その2 解答)）に記入しなさい。

問2（配点 50 点）

AVL 木とは、各節点 v がキーと呼ばれる値 $key[v]$ を持つ二分探索木のうち、任意の節点 v について、その左部分木の高さを $h_L(v)$ 、右部分木の高さを $h_R(v)$ としたとき、

$$|h_L(v) - h_R(v)| \leq 1$$

が成立する平衡二分探索木である。この平衡条件が崩れた場合には、次のいずれかの回転操作（rotation）を行って木を再構成する。

- 右回転（Right Rotation）
- 左回転（Left Rotation）
- 左右回転（Left-Right Rotation）
- 右左回転（Right-Left Rotation）

また、基本的な構造は二分探索木と同じで、「左のキーは親より小さく、右のキーは親より大きい」というルールに従う。

- 1) 10 個のデータ {30, 20, 40, 10, 25, 35, 50, 5, 15, 27} を順に挿入して AVL 木を構築せよ。各挿入のたびに平衡条件を確認し、不平衡が発生した場合には適切な回転操作を行うこと。最終的に得られる木を図示し、その木の高さを答えよ。（ただし、木の高さは根の高さを 0 として数えるものとする。）
- 2) 1) で構築した AVL 木からノード (20) を削除する。削除対象ノード (20) は 2 つの子を持っている。以下の 2 つのケースについて、それぞれ削除操作を行い、再構成後の木を図示せよ。（削除の際に平衡条件が崩れる場合は、適切な回転操作を行って平衡を保つこと。）
 - a. 削除対象ノード (20) を、その左部分木の最大値 (15) で置き換える場合
 - b. 削除対象ノード (20) を、その右部分木の最小値 (25) で置き換える場合
- 3) 1) で構築した AVL 木における各節点のバランス因子 (Balance Factor = 左部分木の高さ $h_L(v)$ - 右部分木の高さ $h_R(v)$) を、すべての節点について表形式で示せ。（ただし、根のみの木の高さは 0、空の部分木の高さは -1 とする。）また、根のキーを 30 としたとき、キー 15, 25, 27 の各ノードまでの経路上のキー列を、探索アルゴリズムの処理順に従ってそれぞれ示せ。

令和8年4月入学（第2回）

山口大学大学院創成科学研究科（工学系）博士前期課程入学試験

受験区分コード54 専門科目（プログラミング(その1)）

以下の問に答え、解答を解答用紙（プログラミング（その1 解答））に記入しなさい。

問1（配点30点）

プログラム1を実行すると、出力1を得た。空欄(A)～(F)を埋め、プログラムを完成しなさい。

・プログラム1

```
#include <stdio.h>
#include <(A)>
#define NUM 4

int main(void)
{
    char array[NUM][20] = {"East", "West", "South", "North"};
    char temp[20];
    int i, j;

    for (i = (B); i < NUM; i++) {
        for (j = (C); j < NUM; j++) {
            if (strcmp(array[i], array[j]) (D) 0) {
                strcpy((E));
                strcpy(array[j], array[i]);
                strcpy((F));
            }
        }
    }

    for (i = 0; i < NUM; i++) {
        printf("%s\n", array[i]);
    }

    return (0);
}
```

・出力1

```
West
South
North
East
```

令和8年4月入学（第2回）

山口大学大学院創成科学研究科（工学系）博士前期課程入学試験

受験区分コード54 専門科目（プログラミング(その2)）

以下の問に答え、解答を解答用紙（プログラミング（その2 解答））に記入しなさい。

問2（配点20点）

プログラム2は最大公約数を求めるプログラムである。空欄(A)～(D)を埋め、プログラムを完成しなさい。

・プログラム2

```
#include <stdio.h>

int gcd_rec(int a, int b)
{
    if (a == 0)
        return (A);
    if (b == 0)
        return (B);

    return gcd_rec(b, a (C) b);
}

int gcd(int x, int y)
{
    int m, n;

    if (x > y) {
        m = x; n = y;
    } else {
        m = y; n = x;
    }

    return gcd_rec((D));
}
```

```
int main(void)
{
    int no1, no2;

    printf("Input1:");
    scanf("%d", &no1);

    printf("Input2:");
    scanf("%d", &no2);

    printf("GCD:%d\n", gcd(no1, no2));

    return (0);
}
```

令和8年4月入学（第2回）
山口大学大学院創成科学研究科（工学系）博士前期課程入学試験
受験区分コード54 専門科目（プログラミング(その3)）

以下の問に答え、解答を解答用紙（プログラミング（その3 解答））に記入しなさい。

問3（配点 20点）

プログラム3はある文字列を別の配列に複製するプログラムである。空欄(A)～(D)を埋め、プログラムを完成しなさい。

・プログラム3

```
#include <stdio.h>
#define MAX 128

int main(void)
{
    char s1[MAX] = "ABCD";
    char s2[MAX] = "EFGH";
    char *p1, *p2;

    /* Copy the string from s2 to s1 */
    p1 = (A);
    p2 = (B);
    while((C) = (D));

    printf("%s\n", s1);

    return (0);
}
```

令和8年4月入学(第2回)
山口大学大学院創成科学研究科(工学系)博士前期課程入学試験
受験区分コード54 専門科目(プログラミング(その4))

以下の問に答え、解答を解答用紙(プログラミング(その4 解答))に記入しなさい。

問4(配点30点)

プログラム4は2点を対角とする長方形の面積を計算するプログラムである。空欄(A)~(F)を埋め、プログラムを完成しなさい。

・プログラム4

```
#include <stdio.h>
#define FABS(x) ((x) < 0.0 ? -(x) : (x))

typedef struct {
    double x, y;
} point;

double area(point *p1, point *p2)
{
    double a;

    /* Calculate the area of a rectangle as height * width */
    a = FABS( (A) - (B) ) * FABS( (C) - (D) );

    return (a);
}

int main(void)
{
    point p1, p2;
    double a;

    p1.x = 1.0;
    p1.y = 2.0;
    p2.x = 2.0;
    p2.y = 1.0;

    a = area( (E), (F) );

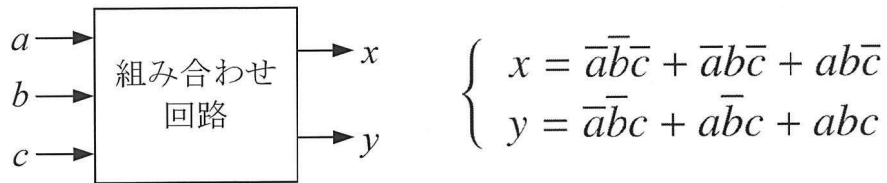
    printf("%f\n", a);

    return (0);
}
```

以下の問に答え、解答を解答用紙（計算機アーキテクチャ(その1 解答)）に記入しなさい。

問1（配点 50点）

以下の図に示すように、入力 a, b, c と出力 x, y が次の論理式を満たす組み合わせ回路を設計する。



- (1-1) 入力 a, b, c に対する出力 x, y の真理値表を作成せよ。
- (1-2) 上記(1-1)の真理値表に基づいて、入力 a, b, c に対する出力 x, y のカルノー図を作成し、それぞれの論理式の最簡形を求めよ。ただし、使用できる論理ゲートは AND, OR, NOT のみとする。
- (1-3) 上記(1-2)で求めた論理式に対応する組み合わせ回路を図示せよ。
- (1-4) 上記(1-3)の回路を NAND ゲートのみを用いて図示せよ。
- (1-5) 上記(1-3)の回路を NOR ゲートのみを用いて図示せよ。

令和8年4月入学（第2回）

山口大学大学院創成科学研究科（工学系）博士前期課程入学試験
 受験区分コード54 専門科目（計算機アーキテクチャ(その2)）

以下の間に答え、解答を解答用紙（計算機アーキテクチャ(その2 解答)）に記入しなさい。

問2（配点50点）

右に示す CASL II のプログラムに関して以下の間に答えなさい。なお、CASL II は仮想ハードウェア COMET II 上で動作するアセンブラ言語であり、数値は16ビットの2進数で表現される。

(2-1) プログラムが終了するまでに、ラベル L0, L1, L2, L3, E1, E2 の行が実行される回数を10進数で答えよ。

(2-2) プログラム終了時に、R から始まる3ワード分の領域に保存される数値を順番に10進数および16進数で答えよ。さらに、S に保存される数値を10進数および16進数で答えよ。

(2-3) 34行から36行を以下のように変更した場合を考える。プログラム終了時に、R から始まる3ワード分の領域に保存される数値を順番に10進数および16進数で答えよ。さらに、S に保存される数値を10進数および16進数で答えよ。

A1 DC 23, 38, 34
 A2 DC 52, 17, 42
 A3 DC 14, 11, 26

(2-4) 34行から36行の A1, A2, A3 を入力、37行から38行の R, S を出力と見なした場合、このプログラムの処理の意味を文章で説明せよ。

[命令形式の例]

LD GRn, address, GRm : GRn ← M(address+GRm)
 ST GRn, address, GRm : GRn → M(address+GRm)
 ADDA GRn, address : GRn ← GRn+M(address)
 SUBA GRn, address : GRn ← GRn-M(address)
 CPA GRn, GRm : GRn = GRm の時に ZF = 1,
 GRn < GRm の時に SF = 1

JMI address : SF = 1 すなわち直前の演算結果が負の時に PR ← address
 JZE address : ZF = 1 すなわち直前の演算結果が零 (zero) の時に PR ← address
 JUMP address : PR ← address
 DC N : 定数 N で指定したデータをメモリに保存
 DS N : N ワード分のメモリ領域を確保

ここで、M(address) は address 番地のメモリの内容、PR は Program Register, ZF は Zero Flag, SF は Sign Flag を表す。

行数	プログラム		
1	MAIN	START	
2		LD	GR0, NUM
3		LD	GR1, =0
4		LD	GR2, =0
5	L0	CPA	GR1, GR0
6		JZE	END
7		LD	GR3, A1, GR1
8		LD	GR4, A2, GR1
9		LD	GR5, A3, GR1
10	L1	CPA	GR3, GR5
11		JMI	L2
12		SUBA	GR3, GR5
13		JUMP	L1
14	L2	CPA	GR4, GR5
15		JMI	L3
16		SUBA	GR4, GR5
17		JUMP	L2
18	L3	CPA	GR3, GR4
19		JMI	E1
20		JZE	E2
21		LD	GR6, =1
22		ADDA	GR2, =1
23		JUMP	NEXT
24	E1	LD	GR6, =-1
25		SUBA	GR2, =1
26		JUMP	NEXT
27	E2	LD	GR6, =0
28	NEXT	ST	GR6, R, GR1
29		ADDA	GR1, =1
30		JUMP	L0
31	END	ST	GR2, S
32		RET	
33	NUM	DC	3
34	A1	DC	2, 3, 3
35	A2	DC	1, 9, 7
36	A3	DC	3, 5, 4
37	R	DS	3
38	S	DS	1
39		END	