

修士課程 社会情報学専攻入学者選抜試験問題

(情報学基礎)

Entrance Examination for Master's Program

(Fundamentals of Informatics)

Department of Social Informatics

平成 25 年 8 月 5 日 10:00～12:00

August 5, 2013 10:00 - 12:00

【注意】

- ・ 試験開始の合図があるまで中を見てはいけない。
- ・ 問題用紙は表紙を含めて 7 枚である。試験開始後、枚数を確認し、落丁または印刷の不鮮明なものがあれば直ちに申し出ること。
- ・ 問題は 5 題である。このうち 3 題を選択し、解答しなさい。
- ・ 解答用紙の表紙に記載されている注意事項についても留意すること。
- ・ 問題 1 題につき、解答用紙 1 枚を使用すること。解答用紙は裏面を使用しても構わないが、使用する場合は裏面に継続することを明記すること。

NOTES

- ・ Do not open the pages before a call of starting.
- ・ This Question Booklet has 7 pages including this front cover.  
After the call of starting, check all pages are in order and notify proctors immediately if missing pages or unclear printings are found.
- ・ There are 5 questions. Choose and answer 3 questions in total.
- ・ Read carefully the notes on the front cover of the Answer Sheets, too.
- ・ Use one sheet for each question. If necessary, the reverse side may be used, stating "Over" at the end of the page.

問題番号 (Number): 1

- (1) ユーザーに 10 個の実数を入力するように求め、その平均値を出力するプログラムになるよう、以下のプログラム中の空欄 (a) および (b) にいくつかの Pascal のコードを書いて完成させよ。

Fill in the blanks (a) and (b) with some Pascal codes to complete the following program, which asks a user to input 10 real numbers and output the average value.

```

program getAverage;
type realarray10 = array[1..10] of real;
var
    A: realarray10;
    i: integer;
    (a)
begin
    i := 1;
    while i <= 10 do
        begin
            writeln('Please input a number. ');
            readln(A[i]);
            i := i + 1;
        end;
        (b)
    end.

```

- (2) ユーザーに 10 個の整数を入力するように求め、クイックソートを用いてそれらを昇順にソートし、結果を出力するプログラムを書け。なお、クイックソートは再帰を用いて実現せよ。用いるプログラミング言語は Pascal, C, Java のいずれかとする。

Write a program that asks a user to input 10 integer numbers, sort them in ascending order using the quicksort algorithm, and output the result. You need to use a recursive procedure for your implementation of quicksort. You can use Pascal, C, or Java as a programming language.

問題番号 (Number) : II

- (1) 個人が作成できる小規模なプログラムに比べて大規模な産業用プログラムの作成が困難なものである事を, 具体的な理由を挙げて論じよ.

Explain why a large-scale industrial program is much more difficult to develop than a program which can be produced by a personal programming providing concrete reasons.

- (2) 大規模な産業用プログラム作成における困難を克服する方法について論じよ.

Explain how to overcome difficulties in developing a large-scale industrial program.

問題番号 (Number) : III

以下の生成規則 R1, R2, R3, R4 と R5 を用いて, 次の文

$$U := (X + Y) * Z$$

を P88 マシンのアセンブリ言語に翻訳せよ. P88 マシンの命令セットは表 1 に示す.

Use the production rules R1, R2, R3, R4 and R5 shown below to translate the following statement into assembly language of P88 Machine. The instructions of P88 machine are shown in Table 1.

$$U := (X + Y) * Z$$

#### 構文(Syntax)

R1:  $\langle i \rangle_j \rightarrow w$

R2:  $\langle e \rangle_i \rightarrow \langle i \rangle_j$

R3:  $\langle s \rangle_k \rightarrow \langle i \rangle_j := \langle e \rangle_i$

R4:  $\langle e \rangle_i \rightarrow (\langle e \rangle_j + \langle e \rangle_k)$

R5:  $\langle e \rangle_i \rightarrow (\langle e \rangle_j * \langle e \rangle_k)$

#### 意味(Semantics)

$M(\langle i \rangle_j) = w$

$M(\langle e \rangle_i) = M(\langle i \rangle_j)$   
 $code(\langle e \rangle_i) = \text{nothing}$

$code(\langle s \rangle_k) = code(\langle e \rangle_i)$   
 COPY AX, M( $\langle e \rangle_i$ )  
 COPY M( $\langle i \rangle_j$ ), AX

$M(\langle e \rangle_i) = \text{createname}$   
 $code(\langle e \rangle_i) = code(\langle e \rangle_j)$   
 $code(\langle e \rangle_k)$   
 COPY AX, M( $\langle e \rangle_j$ )  
 ADD AX, M( $\langle e \rangle_k$ )  
 COPY M( $\langle e \rangle_i$ ), AX

$M(\langle e \rangle_i) = \text{createname}$   
 $code(\langle e \rangle_i) = code(\langle e \rangle_j)$   
 $code(\langle e \rangle_k)$   
 COPY AX, M( $\langle e \rangle_j$ )  
 MUL AX, M( $\langle e \rangle_k$ )  
 COPY M( $\langle e \rangle_i$ ), AX

ただし,  $\langle i \rangle_j$ ,  $\langle e \rangle_i$ ,  $\langle e \rangle_j$ ,  $\langle e \rangle_k$ ,  $\langle s \rangle_k$  は文法変数である.  $w$  は識別子の文字列である.  $M$  は意味関数である.

where,  $\langle i \rangle_j$ ,  $\langle e \rangle_i$ ,  $\langle e \rangle_j$ ,  $\langle e \rangle_k$ , and  $\langle s \rangle_k$  are grammar variables.

$w$  is some identifier string. The notation  $M$  stands for the concept of "meaning".

表 1 命令セット (TABLE 1 Instruction Set)

命令 Instruction	フォーマット Format	アクション Action
Copy from mem	COPY AX, mem	AX:=mem
Copy to mem	COPY mem, AX	mem:=AX
Add	ADD AX, mem	AX:=AX+mem
Subtract	SUB AX, mem	AX:=AX-mem
Multiply	MUL AX, mem	AX:=AX*mem
Divide	DIV AX, mem	AX:=AX div mem
Compare	CMP AX, mem	If AX < mem then CF:=B else CF:=NB
Jump	JMP lab1	Go to the instruction with label lab1
Jump if not below	JNB lab1	Go to the instruction with label lab1 if CF=NB. Otherwise go to next instruction
Jump if below	JB lab1	Go to the instruction with label lab1 if CF=B. Otherwise go to next instruction
Input	IN AX	Input an integer into register AX
Output	OUT AX	Output an integer from register AX

## 問題番号 (Number) : IV

計算の実現可能性の観点に基づいた現在のコンピュータ科学の限界に関連して、以下の問いに答えよ。

Answer the following questions about the current limitation of computer science concerning the possibilities of intractable calculations.

- (1) 答えを得るために膨大な時間を要するため、計算が実現できない問題がある。計算を高速化する方法として並列計算の技術があるにも関わらず、この問題を克服できないのは何故か論じよ。

It is practically impossible for some programs to get the answer because of their too long execution time. Although parallel computation is a way to speed up a calculation, it has not overcome the difficulty in executing such programs. Explain its reason.

- (2) いかなるコンピュータを使っても問題解決ができない、いわゆる計算不能であるとはどういう事か。計算困難との違いを含め説明せよ。

Explain noncomputability which means we cannot solve the problem with any high-speed computer. Note that you must explain differences between "noncomputability" and "intractable computation".

- (3) 計算困難な問題の例として良く取り上げられる巡回セールスマン問題を多項式時間内で解く方法が発見された場合の影響について説明せよ。

Traveling Salesperson Problem (TSP) is a typical example of intractable computations. Explain the impact of finding the solution of TSP.

問題番号(Number):V

以下の用語から四つを選び, 説明せよ.

- ・コンピュータサイエンス
- ・CPU
- ・命令取り出しー実行サイクル
- ・並列計算
- ・知識表現におけるフレーム
- ・加算回路

Choose four terms from the following list and explain them.

- ・Computer science
- ・CPU
- ・Fetch-execute cycle
- ・Parallel computation
- ・Frame in knowledge representation
- ・Adder circuit