

コンピュータサイエンス第 2—期末試験 CS4b—

答案用紙は各問ごとに 1 枚使用して書くこと。

答案用紙には各枚ごとに学籍番号と氏名を書くこと。

問 1 . (配点 10 点)

つぎの問に答えよ . 計算の過程も解答用紙に残すこと . $(n)_m$ は n が m 進表記であることを表すものとする .

- (1) $(0.1)_{10}$ を 32 bits の浮動小数点数に変換せよ . 符号に 1 bit , 指数に 8 bits , 仮数に 23 bits とする . ただし , 規格に従って指数部は下駄をはかせること .
- (2) $(5.005)_{10} + (1.00003)_{10}$ を 10 進で 5 桁まで記憶できるコンピュータで計算したときの丸め誤差を求めよ (2 進数に変換せず , 10 進のままで考えてよい) .
- (3) つぎの無限級数の和と第 n 項までの和の差の絶対値は第 $n+1$ 項の絶対値より小さくなる . $x = 1.0$ として第 5 項で打ち切ったときの打ち切り誤差の絶対値の上限を求めよ . ここで $n = 0, 1, \dots$ とする .

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} \cdots + (-1)^n \frac{x^{2n+1}}{(2n+1)!} \cdots$$

問2 . (配点 10 点)

プログラミング言語 Ruby で書かれたプログラムの一部を以下に示す . このプログラムについてつぎの問いに答えよ .

```
1 def min_el(dis,rest)
2   tmp=rest.first
3   rest.each do |v|
4     if(dis[tmp]>dis[v])
5       tmp=v
6     end
7   end
8   return(tmp)
9 end
10 def min(a,b)
11   if(a<b)
12     return(a)
13   else
14     return(b)
15   end
16 end
17 def dijkstra(start, destination, graph, p)
18   d=Array.new(28)
19   s=Array.new(28)
20   rest=[0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27]
21   for i in 0..27
22     d[i]=graph[start][i]
23   end
24   s.push(start)
25   rest.delete(start)
26   n=rest.length
27   i=0
28   while i<n
29     w=min_el(d,rest)
30     s.push(w)
31     rest.delete(w)
32     rest.each do |v|
33       if(d[w]+graph[w][v]<d[v])
34         d[v]=d[w]+graph[w][v]
35         (1)
36       end
37     end
38     i=i+1
39   end
40   return(p)
41 end
```

- (1) p は最短経路を保持するための配列である . これをもちいて (1) の中を完成させよ .
- (2) rest.delete(e) は配列 rest から要素 e を削除した配列を , rest.each do |v| は配列 rest の各要素 v についての繰り返しを意味している . このとき , グラフのノード数 n として 17 行-41 行までの時間計算量を求めよ .

問3 . (配点 10 点)

プログラミング言語 Ruby で書かれたプログラムの一部を以下に示す . このプログラムについてつぎの問いに答えよ .

```
1 def eps_m ()
2   epsilon=1.0
3   old=0.0
4   prod=0.0
5   cnt=0
6   while (prod!=1.0)
7     old = epsilon
8     cnt=cnt+1
9     epsilon=epsilon/2.0
10    prod=epsilon+1.0
11  end
12  return(old)
13 end
14 #
15 def f_iter (guess,n,eps,previous)
16   if is_enough(guess,eps,previous) then
17     return(guess)
18   else
19     return(sqrt_iter(improve(guess,n),n,eps,guess))
20   end
21 end
22 def improve (guess,n)
23   return((guess+(n/guess))/2.0)
24 end
25 def is_enough (guess,eps,previous)
26   return(abs(previous-guess)<(2.0*eps))
27 end
28 def abs (x)
29   if x<0 then
30     return(-x)
31   else
32     return(x)
33   end
34 end
35 def f1 (n,eps)
36   return(f_iter(1.0,n,(2*eps),0.0))
37 end
38 def f (n)
39   print(f1(n,eps_m()),"\n")
40 end
```

- (1) 漸化式で表せるものは自然に再帰プログラムに表せる . `f_iter()` を漸化式の形で示せ .
- (2) `is_enough` で数列は収束したと判定している . このときの収束判定条件を示せ .
- (3) `eps_m()` は何を求めている関数か述べよ .
- (4) このプログラムを用いて `f(2.0)` を計算したときの各繰り返しでの `guess` の値を示せ . ただし , 10 進表記で 9 桁まで記憶できると仮定する .
- (5) 自然数 n に対して , $f(n)$ は , どのような関数か述べよ .