令和７年度（２０２５年度）
大分大学理工学部総合型選抜

知能情報システムプログラム　筆記試験

検査時間　60 分（9:00～10:00）

注意事項

１．試験開始の合図があるまで，この問題冊子の中を見てはいけません。
２．受験番号を解答用紙の所定の欄に記入してください。
３．解答は解答用紙の指定された解答欄に記入してください。
４．試験時間中に問題冊子及び解答用紙の印刷不鮮明，ページの落丁及び
　　汚損等がある場合は，手を挙げて監督者に知らせてください。

1 フィボナッチ数列は漸化式 $a_n = a_{n-1} + a_{n-2}$ $(n \geqq 2,\ a_0 = 0,\ a_1 = 1)$ で定義される。リスト 1 はフィボナッチ数列 (最大で a_{46}) を求める C 言語のプログラムである。リスト 1 では，a_1 から a_n の値を出力する。フィボナッチ数列 a_n の n はキーボードから 2 以上 47 未満の整数値を入力して与えるものとする。図 1 および図 2 に n として 7 と 15 を与えた際の実行結果を示している。図 1 と図 2 の下線部は，キーボードからの入力を表している。

実行結果と同じ出力結果が得られるように，リスト 1 の空白部【ア】から【カ】を埋めてプログラムを完成させなさい。

リスト 1　フィボナッチ数列を求めるプログラム

```
1  #include<stdio.h>
2
3  int main(){
4    int i, n, a[47];
5    a[0] = 【 ア 】;
6    a[1] = 【 イ 】;
7
8    printf("n: 2以上47未満の整数を入力 = ");
9    scanf("%d", 【 ウ 】);
10
11   if(n > 46){
12     n = 46;
13   }
14
15   for(i = 【 エ 】; i <= n; i++){
16     a[i] = 【 オ 】;
17   }
18
19   for(i = 0; i < n; i++){
20     printf("a[%d] = %d\n", 【 カ 】, a[【 カ 】]);
21   }
22   return 0;
23 }
```

```
n:  2以上47未満の整数を入力 = 7
a[1] = 1
a[2] = 1
a[3] = 2
a[4] = 3
a[5] = 5
a[6] = 8
a[7] = 13
```

図 1　実行結果例 (1)

```
n:  2以上47未満の整数を入力 = 15
a[1] = 1
a[2] = 1
a[3] = 2
··· (途中省略) ···
a[13] = 233
a[14] = 377
a[15] = 610
```

図 2　実行結果例 (2)

2 リスト 2 は，与えられた引数からある計算をして順番に求めた数値を出力する C 言語で作成した関数である。リスト 2 に関して，以下の問いに答えなさい。

(1) リスト 2 に示した関数 c の引数 n に 3 を与えたときに得られる出力結果を答えなさい。

(2) リスト 2 に示したプログラムは再帰呼び出しを用いて繰り返し処理を実現している。リスト 2 の繰り返し処理における終了条件を日本語で答えなさい。

(3) リスト 2 の再帰呼び出しを用いた繰り返し処理を while 文を用いて書き直した C 言語のプログラムをリスト 3 に示す。リスト 3 の関数 c の引数 n に 3 を与えたとき，リスト 2 と同じ実行結果が得られるようにリスト 3 の空白部【ア】から【エ】を埋めてプログラムを完成させなさい。ただし，空白部【イ】は選択肢 (a) から (d) の中から適切なものをひとつ選びなさい。

リスト 2　ある計算をするプログラム

```
1  #include<stdio.h>
2
3  void c(int n){
4    printf("%d ", n);
5
6    if((n % 2 == 1) && (n > 1)){
7      printf(" -> ");
8      c(3*n+1);
9    }else if(n % 2 == 0){
10     printf(" -> ");
11     c(n/2);
12   }
13 }
```

リスト 3　リスト 2 を while 文で書き直したプログラム

```
1  #include<stdio.h>
2
3  void c(int n){
4
5    while( 【 ア 】){
6      printf("%d ", n);
7      if(n == 1){
8        printf("\n");
9        【 イ 】;
10     }
11     if((n % 2) == 1){
12       printf(" -> ");
13       【 ウ 】;
14     }else if((n % 2) == 0){
15       printf(" -> ");
16       【 エ 】;
17     }
18   }
19 }
```

空白部【イ】の選択肢

(a) continue　(b) goto　(c) break　(d) auto

3 以下のデータ圧縮法に関する文章を読んで，次の問いに答えなさい。

データを圧縮する基礎的な方法のひとつにランレングス法がある。ランレングス法は同一のデータが連続する場合にデータ 1 つとその個数を示すことでデータを圧縮する方法である。例えば，文字列 “AAAAAABBBC” は，A6B3C1 と圧縮される。これは，A が 6 個連続，B が 3 個連続，C が 1 個連続していることを表している。

(1) 図 3 に示した白黒画像をランレングス法で圧縮する。図 3 は 7×7 のピクセルで大文字 E を表現している。図 4 のように，図 3 における黒で塗りつぶしたピクセルを B，白で塗りつぶしたピクセルを W で表現したとき，ランレングス法で圧縮した結果を答えなさい。図 4 は 7×7 のピクセルで表しているが，開始位置の B を先頭に矢印の順に連続した 49 文字の文字列として考えることとする。

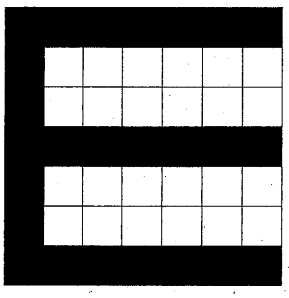

図 3　白黒画像 (7×7 ピクセル) で表した大文字 E

開始位置

B	B	B	B	B	B	B
B	W	W	W	W	W	W
B	W	W	W	W	W	W
B	B	B	B	B	B	B
B	W	W	W	W	W	W
B	W	W	W	W	W	W
B	B	B	B	B	B	B

図 4　図 3 の白黒を W と B で表現

(2) ランレングス法でデータ圧縮する C 言語のプログラムをリスト 4(次ページ) に示す。リスト 4 のプログラムにおいて，黒を表す B の代わりに 1，白を表す W の代わりに 0 を用いて配列 image で図 3 の白黒画像を表現している。以下に示す処理内容に従い動作するように，リスト 4 の空白部【ア】から【エ】を埋めてプログラムを完成させなさい。

処理内容

配列 image の i 番目の要素の文字を基準として，i+1 番目の要素の文字と比較し，一致すれば同じ文字が連続していることになる。連続していれば，比較対象を i+2 番目の要素の文字としていく。このように，連続している場合は，比較対象とした文字の要素番号を 1 つずつ増やしていくようにする。

一方，基準となる文字と比較対象の文字が一致しなかった場合，配列 cimage の偶数番目 (loc) の要素に基準となる文字，奇数番目 (loc+1) の要素にこれまで連続した文字数を代入する。今，比較対象としていた要素番号を次に基準となる要素番号とする。

(3) ランレングス法でデータ圧縮効率が最も悪い白黒画像を日本語で説明しなさい。

リスト４　データ量圧縮プログラム

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

int main(){
  int i = 0;
  int k;
  int loc;
  int count = 1;  /* 連続する個数を数え上げる変数 */
  /* cimage: ランレングス法により圧縮した結果を格納する配列
     (偶数番目の要素: 文字, 奇数番目の要素: 連続する文字数) */
  char cimage[49*2];
  memset(cimage, -1, sizeof(cimage)); /* 配列cimageすべての要素を-1で初期化 */
  int image[49] = {1, 1, 1, 1, 1, 1, 1,
                   1, 0, 0, 0, 0, 0, 0,
                   1, 0, 0, 0, 0, 0, 0,
                   1, 1, 1, 1, 1, 1, 1,
                   1, 0, 0, 0, 0, 0, 0,
                   1, 0, 0, 0, 0, 0, 0,
                   1, 1, 1, 1, 1, 1, 1}; /* 配列imageは図3の白黒画像を表現 */

  loc = i;
  k = i+1;
  while(i < 49){
    if(image[i] == image[k]){
    /* 基準となる文字が連続している場合の処理 */
      count++;
      k++;
    }else{
    /* 基準となる文字と比較対象の文字を比較して一致していない場合の処理 */
    /* 配列cimageに文字と連続する文字数を格納 */
      cimage[loc] = 【 ア 】;
      cimage[loc+1] = 【 イ 】;
      loc += 2;
      count = 【 ウ 】;
      i = 【 エ 】;
    }
  }

  for(i = 0; i < 98; i+=2){
    if(cimage[i] != -1){
      printf("%d:%d\n", cimage[i], cimage[i+1]);
    }else{
      break;
    }
  }
  return 0;
}
```