

## 2015 年度プログラミング C 期末試験 解答例(102 点満点)

### 問題 1 (各1点x13=13点)

- ( 1) int
- ( 2) char \*
- ( 3) FILE
- ( 4) argc
- ( 5) 3
- ( 6) fprintf
- ( 7) stderr
- ( 8) fopen(argv[1], "r")
- ( 9) NULL
- (10) fopen(argv[2], "w")
- (11) fgetc(fpin)
- (12) putchar(c)  
printf("%c",c) や fputc(c,stdout), fprintf(stdout, "%c", c) でも可
- (13) fclose

### 問題 2 (各2点x10=20点)

- ( 1) arr  
    &arr[0] も可
- ( 2) arr + N  
    &arr[0]+N や &arr[N] も可
- ( 3) p++
- ( 4) p < arr + N - 1  
    p <= arr + N - 2 も可
- ( 5) p
- ( 6) \*q
- ( 7) \*maxptr
- ( 8) \*p
- ( 9) \*maxptr
- (10) tmp

## 問題1-2講評

まず、全体を通してですが、あまり点数をとれていた人が少なく、大体の人が半分くらいの出来栄でした。

### 問題 1

(2)では、char のあとに必要な \*のつけ忘れがとても多かったです。

(5)では、2 にしている人が何人かいました。コマンドライン引数はコマンド名自身も含むので気をつけてください。

(6)は、fprintf ではなく、printf にしている人がいました。

(8), (10)ではダブルクォーテーション (") ではなく、シングルクォーテーション (') で囲っている人、

(10)で読み込みモードで fopen している人、argv の index 番号を間違えている人などがいました。

(11)では fgetc ではなく、fscanf や scanf を使ってファイルを読み取っている人が多かったです。また、fgetc と (12)の putchar で引数の内容を間違えている人も頻繁に見受けられました。

(13)では fclose ではなく、free を使っている人がいました。

### 問題 2

全体的に\*、&などのつけるところ、つけないところの間違いが多かったです。ポインタと値の実態、アドレスの考えがうまく定着してないように感じました。

(4)では  $p < arr + N - 1$  と書くところを右辺のみ書いている人が見受けられました。

(10)では tmp を temp とスペルのミスをしている人が何人かいました。

スペルのミスについては問題 1 でもいましたが、問題 1 では不正解として、問題 2 では 1 点の減点として採点を行いました。

### 問題 3 (各1点x20=20点)

- ( 1) typedef
- ( 2) p1
- ( 3) p2
- (2) と (3) は順不同
- ( 4) iscross(\*p1, \*p2)  
引数の順番は入れ替わっていても可
- ( 5) ->x
- ( 6) ->y
- ( 7) ->q
- ( 8) c%10
- ( 9) c/10
- (10) malloc
- (11) &p->x  
  &(\*p).x も可
- (12) &p->y  
  &(\*p).y も可
- (13) p
- (14) 3
- (15) 4
- (16) p->q  
  (\*p).q も可
- (17) quad
- (18) xyq a, xyq b  
  xyq b, xyq a も可
- (19) fabs(a.q - b.q)  
  a.q-b.q==2 || b.q-a.q でも可
- (20) a.q + b.q == 5  
  その他どのような方法でも 1, 4 と 2, 3 になれば可

### 問題 4 (4点+9点=13点)

#### 問題 4-1 (各 2 点 x2=4 点)

- ( 1) normal.h
- ( 2) gcc q4-1.c -DWINDOWS (-DWINDOWS=値 でも可)

#### 問題 4-2 (各 1 点 x9=9 点)

	(a,b)	(a+1,b)	(a,b)*2
func1	: 13	14 X	22 X
func2	: 13	18	22 X
func3	: 13	18	26

## 問題3-4講評

### 問題 3

- (1) 比較的できていたが、`typedef`, `typedf`, `Typedef` 等のスペルミスが少しあった。
  - (2) (3) 必要のないアスタリスクをつけている場合が少なからずあった。
  - (4) アスタリスク (\*) をつけ忘れたり、逆にアンパサンド (&) をつけてしまう人が散見された。
  - (5) (6) (7) アロー演算子ではなくドットを書く人も一定数みられた。また (7) では `x, y` に続いて `q` ではなく `z` を書いてしまった人がいました。
  - (8) (9) 解答は剰余算、除算を想定していましたが、`c` の値が、0, 1, 10, 11 しかないことを利用するなどした様々なバリエーションの解答がありました。正しく動作するかということを念頭に採点しましたが、条件式の優先順位を理解していないのか論理がおかしい解答も見られました。
  - (10) 出来が良かった
  - (11) (12) 入力にポインタを使うので `scanf` の `&` を忘れる人が多く確認された。
  - (13) 必要のないアスタリスクをつける人が多くみられた。
  - (14) (15) (14) (15) の解答を逆に答える人がいました。
  - (16) (17) 出来が良かった
  - (18) 正答率の高いものの、`xyq` ではなく `xyz` と答える人が一部確認された。
  - (19) 正誤を問わず回答のバリエーションが非常に豊富だった。
- 模範解答の `fabs` ( or `abs`) が一番すっきりしていると思いますが、差を2乗して平方根をとったり、すべての場合について記述した例などがありました。

- (20) 足して5かどうかを判定するのか一番簡単と思いますが、すべての場合について記述した解答がありました。また、2変数の差から条件を求めようとした例もありましたが、ほとんどの場合、他の組み合わせも成り立つのでNGでした。

### 問題 4

- 4-1 (1) 比較的正答率が高かった。
- 4-1 (2) `q4-1.c` を書き忘れたり、`windows.h` をコンパイルしようとした解答がありました。
- 4-2 計算結果ではなく計算式で答える人がいた他、`X` を付け忘れている人も多くいたようです。また、どこかで計算間違いをしてしまった人も多く、全問正解した人はほとんどいませんでした。

## 問題 5 (各2点x8=16点)

- ( 1) `i>n-k ; i--`  
`i>=n-k+1 ; i-` も可
- ( 2) `r*=i`
- ( 3) `i>1 ; i--`  
`i>0 ; i--` も可 (他に `i>=2`, `i>=1` も可)
- ( 4) `r/=i`
- ( 5) `k==0 || k==n`  
`!(k>0 && k<n)` も可
- ( 6) `combi_r(n-1,k-1) + combi_r(n-1,k)`
- ( 7) `int j, int *p`  
`*p` は `p[]` 等も可
- ( 8) `p[i] = p[i-1]*i`  
`*(p+i) = *(p+i-1)*i` も可

## 問題 6 (各2点x10=20点)

- ( 1) `typedef struct node`
- ( 2) `make_1node( 0, NULL )`
- ( 3) `n=head->next; n!=NULL`
- ( 4) `sizeof(struct node)`
- ( 5) `n->next->key`
- ( 6) `n->next = nnew`
- ( 7) `1`
- ( 8) `n->next`
- ( 9) `ndel == NULL`
- (10) `ndel->next`  
`n->next->next` でも同じ

## 問題5-6講評

### 問題 5

問題文に出ている `nCk` の数式の計算をプログラムにするだけの問題でしたが、意外に正答率が低めでした。`for` ループの使い方や関数の再帰を利用した計算など、出来なかった人はよく復習しておいてほしいと思います。

(1-2) なぜか、特に出来が良くなかったようです。デクリメント `i--` を使ったループに慣れていないのかもかもしれませんが、落ち着いて考えれば難しくない所です。

なお、ループ継続／終了の判定では、比較の演算子に `>` を使うか `=` を使うかで右辺の値が 1 違ってきますので、注意してください。

(5) は比較的に出来ていましたが、OR の演算子を `&&` とか、`|` にしてしまう誤答が見られました。

(8) では、`p[i]*=i` のような、階乗の意味がよく分かっていないような誤答もあったようです。

### 問題 6

半分以上できている人は少なかったですが、例年よりは少し平均点は高めのようなようです。

(1) 問題文側に `*` があるのに、さらに `*` をつけてしまっている人や、一単語抜けてしまっている答案がありました。

(2) `NULL` ポインタを整数 `0` にしてしまっていたり、変数宣言時の初期化と混同して `{0,0}` などとしている誤答が目立ちました。

(3) リスト操作で重要なループの処理ですが、出来が良くないです。コメントに `head` 以外を出力する、と明記してありますので、`n=head->next` から始まり、ループ継続条件は `n!=NULL` でなければなりません。

(4) `sizeof(NodePointer)` という誤答が目立ちました。これだとポインタのサイズ (Solaris で 4 バイト、Mac で 8 バイト) になってしまいます。

(9) `n->next==NULL` としていた人が多いのですが、このリストではデータが昇順に並んでいて、リストの末尾まで探さなくても検索するデータがリストに無いことが分かるため、途中で検索のループを `break` で脱出する仕組みになっています。この場合 `n->next` は `NULL` になりません。