

# 同人技術

(60点  
30分)

## I 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 問題は第1問～第3問の3題あり、すべて必須問題です。
- 3 試験中に問題冊子の不鮮明、ページの落丁・乱丁及び解答用紙の汚れ等に気づいた場合は、手を挙げて監督者に知らせなさい。
- 4 この冊子は同人技術の問題です。同人グラフィック、同人音楽を選択したのにこの冊子が届いているものは、手を挙げて監督者に知らせなさい。
- 5 解答用紙には解答欄以外に次の記入欄があるので、監督者の指示に従って、それぞれ正しく記入し、マークしなさい。
  - ① 受験番号欄  
受験番号（数字及び英字）を記入し、さらにその下のマーク欄にマークしなさい。正しくマークされていない場合は、採点できないことがあります。
  - ② 氏名欄、試験場コード欄  
氏名・フリガナ及び試験場コード（数字）を記入しなさい。
- 6 問題冊子の余白等は適宜利用してよいが、どのページも切り離してはいけません。
- 7 試験終了後、問題冊子は持ち帰りなさい。

## II 解答上の注意

- 1 問題中のプログラムは、とくに指定のない限り C++言語によって書かれ、規格は JIS X 3014:2003 『プログラム言語 C++』に準拠するものとします。
- 2 プログラム中の文字は、サンプルにしたがって正しく判別するようにしなさい。

数字の 1	1
大文字の I (アイ)	I
小文字の l (エル)	l
数字の 0	0
大文字の O (オー)	O
小文字の o (オー)	o





# 同人技術

( 全 問 必 答 )

第1問 次の文章を読み、下の問いに答えよ。(配点 20)

私たち同人開発者が活躍する場として、さまざまな同人誌即売会があるが、ここでは、その中でコミックマーケットについて考えてみよう。コミックマーケットは1年に2回、夏と冬に、その両方が①東京ビッグサイトで行われる。ここ数年は夏、冬とも□1□日間開催が一般的である。参加区分としては、サークル参加者、□2□、スタッフ参加者、一般参加者の4つがあり、開発者がサークルとして参加するとき、これはサークル参加者としての参加となる。一般参加者というのは、事前の登録等を必要とせず、当日主に②作品の購入をする来場者のことである。この中には、コミックマーケットの□3□という理念を理解しない迷惑な振る舞いをしたり、本来禁止である③徹夜をして近隣住民に迷惑をかけたという者もあり、問題となっている。スタッフ参加者は、コミックマーケット運営を行うボランティア参加者である。当日は列の整理等を行う。スタッフ参加をするには、東京で行われる事前の□4□に複数回参加しなければならない。

コミックマーケットでは、④ジャンルごとに開催日を区分している。そのために「ジャンルコード」が規定されており、例えば「デジタル(その他)」「同人ソフト」「東方Project」はいずれも2日目に行われるため、同人ゲームや音楽、映像作品を求める人はこの日に集まる傾向にある。このような細かい区分がなされているところからも、コミックマーケットの規模の大きさ、そして運営の苦勞をうかがい知ることができる。

このように、コミックマーケットはたくさんの参加者の力で作り上げられるものである。単にコミックマーケットを貸し店舗のように見るのではなく、自らもイベント作りに参加し、また他の参加者の活動に感謝しつつ、楽しんで参加するのが重要だといえるだろう。

問1 文章中の  に入れるのに最も適当な数字を ① ~ ⑨ のうちから一つ選べ。

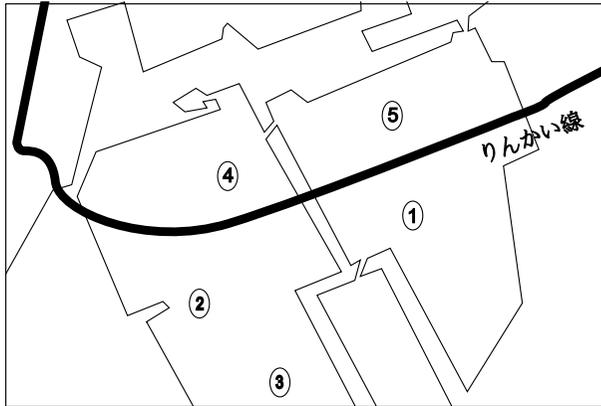
問2 文章中の  ・  ・  に入れるのに最も適当なものを、次のそれぞれの ① ~ ④ のうちから一つずつ選べ。

{ ① 招待参加者 ② 特別参加者  
③ 企業参加者 ④ 委託参加者

{ ① お客様は「神様」です ② 「お客様」は存在しない  
③ 「神は死んだ」 ④ 「サークルのための」コミケ

{ ① 拡大準備集会 ② 大発会  
③ コミケ準備集会 ④ 全体準備集会

問3 下線部①に関連して、東京ビッグサイトの所在地として適当な場所を、次の地図の ① ~ ⑤ のうちから一つ選べ。



問4 下線部②に関連して、作品を購入する際のマナーとして適当でないものを、次の ① ~ ④ のうちから一つ選べ。

- ① 購入する際に、周りがすいていたので、作者に過去の作品の感想を伝えた。
- ② 作者の荷物がかさばらないよう、コミケ終盤に購入するときに100円の商品を1000円札で買い、900円のおつりをもらった。
- ③ 購入する際に作者に辛子明太子の差し入れをした。
- ④ 列の最後尾についたので、最後尾札を持った。

問5 下線部③に関連して、徹夜をしてはいけない理由として適当でないものを、次の①～④のうちから一つ選べ。

- ① 夜の治安は必ずしも良くはなく、盗難、恐喝、性犯罪等に巻き込まれる危険があるから。
- ② 徹夜をするほどの参加者は、極度に熱心な何かのファンであるため、現実と空想の区別がつかなくなってしまう恐れがあるから。
- ③ 徹夜の長時間の野外活動は体力を消耗させ、コミケの苛酷な環境の下で具合を悪くし、最悪救護のスタッフのお世話になってしまう恐れがあるから。
- ④ 徹夜組の存在で発生する警備活動に多大な費用、労力がかかり、コミケの運営を難しくするから。

問6 下線部④に関連して、C78において、次のジャンルの組み合わせのうち、全てのジャンルが同じ日に行われる組み合わせでないものを、次の①～④のうちから一つ選べ。

- ① 「ガンダム」「スポーツ」「ゲーム(RPG)」
- ② 「ギャルゲー」「鉄道・旅行・メカミリ」「特撮・SF・ファンタジー」
- ③ 「歴史」「創作」「Leaf&Key」
- ④ 「FC(少年)」「FC(少女)」「FC(青年)」



第2問 次の説明を読んで、下の問いに答えよ。(配点 20)

コンピュータの性能が向上しつつある今においても、高速化の重要性が失われたわけではない。ここでは、コンピュータに近づくための基本である、ビット演算についてみてみよう。今、アルファチャンネルを持った画像データの処理のコードを書いている。また、このアルファチャンネルは、128 以上が不透明、128 未満が透明という約束にしてあり、色情報の上 2 バイトに格納してあるものとしよう。この状況で、unsigned int 型の素子の色情報を引数に、その素子が不透明なら true を返す bool 型関数 isTransparent を作りたい。これを単純に書くと次のようになる。なお、この環境では unsigned int 型は 32 ビットの情報を保持できるものとする。

```
bool isTransparent(unsigned int color){
    unsigned int alpha = (color & 0xff000000) >> 9;
    return alpha >= 10;
}
```

しかしこれはまだ冗長である。一旦、このコードを次のように分解してみよう。

```
bool isTransparent(unsigned int color){
    unsigned int a = color & 0xff000000
    a >>= 9
    return a >= 10;
}
```

ここで、4 行目の式の両辺に 9 ビットの左シフトをかけると、

```
bool isTransparent(unsigned int color){
    unsigned int a = color & 0xff000000
    return a >= 0x11;
}
```

このようになる。さらに、128 以上 255 以下の数は、2 進法にすれば必ず 12 の桁が 1 になる。つまり、12 の桁が 1 であることと、その数が 128 以上であることは同値である。これを利用して 255 以下の数が 128 以上であるかを判定するには、(判定する数) & 0x13 を考えればよく、アルファチャンネルは 9 ビットシフトしたところにあるのだから、13 を 9 ビット左シフトした、0x14 との論理積を取ればよい。したがって、

```
bool isTransparent(unsigned int color){  
    return color & 0x[ 14 ];  
}
```

ここまでコードを簡略化することができた。

問1 [ 9 ] · [ 10 ] · [ 11 ] · [ 12 ] · [ 13 ] · [ 14 ] に当てはまる数字として適当なものを、次の選択肢の中からそれぞれ一つ選べ。

- ① 6
- ② 24
- ③ 40
- ④ 60
- ⑤ 64
- ⑥ 80
- ⑦ 128
- ⑧ 4000000
- ⑨ 6000000
- ⑩ 8000000

第3問 次の記事を読んで、下の問いに答えよ。(配点 20)

## New OpenGL 4.0 aims to match DirectX 11

Aiming to keep pace with Microsoft and advance the computing frontier, the group behind OpenGL has announced a new version of its interface designed to make advanced graphics easier for programmers to handle.

OpenGL 4.0 adds more support for <sup>①</sup>using a graphics processing unit (GPU) for other computing chores and for tessellation, which subdivides a region on a graphics object into many smaller patches for more detailed imagery. The technology got its start as a graphics library at pioneering Silicon Graphics but has grown into a standard that works on many different computer systems and overseen by the Khronos Group.

”OpenGL 4.0 exposes the same level of capability of GPUs as DirectX 11,” the competing interface from Microsoft, Khronos said in a presentation. The company announced the new standard, along with the 3.3 update to the earlier version, at the Game Developers Conference on Thursday.

OpenGL 4.0 is compatible with a technology called OpenCL that’s designed to let graphics chips handle many chores that otherwise would be the work of the main processor. Graphics chips aren’t suited to all tasks, but they can help with some computationally intense work such as videogame physics. Graphics chips are steadily growing more powerful and widespread since modern operating systems, not just shoot-em-up videogames, now require 3D graphics abilities.

If you want to plunge into the details, here’s how Khronos describes the new OpenGL 4.0 abilities:

- two new shader stages that enable the GPU to offload geometry <sup>②</sup>tessellation from the CPU;
- per-sample fragment shaders and programmable fragment shader input positions for increased rendering quality and anti-aliasing flexibility;
- drawing of data generated by OpenGL, or external APIs such as OpenCL, without CPU intervention;
- shader subroutines for significantly increased programming flexibility;
- separation of texture state and texture data through the addition of a new object type called sampler objects;
- <sup>③</sup>64-bit double precision floating point shader operations and inputs/outputs for increased rendering accuracy and quality;
- performance improvements, including instanced geometry shaders, instanced arrays, and a new timer query.

OpenGL also has variations called OpenGL ES for mobile devices including Apple's iPhone and WebGL for Web browser applications.

出典:DeepTech [http://news.cnet.com/8301-30685\\_3-20000277-264.html](http://news.cnet.com/8301-30685_3-20000277-264.html)

問 1 下線部①に関連して、このような技術のことを一般になんと言うか。適切なものを

① ~ ④ から一つ選べ。

- ① GGPU
- ② GNU
- ③ GNGPU
- ④ GPGPU

問 2 下線部②に関連して、テセレーションとは何かを説明したものとして適切なものを

① ~ ④ から一つ選べ。

- ① よりきめ細かな画像を実現するためにグラフィックスオブジェクト上の領域を多数の小さな区画に分割する処理
- ② より滑らかな映像を実現するために画面のバッファを複数用意する処理
- ③ よりきめ細かな画像を実現するために小さな画像を複数のコンピュータに演算を分散する処理
- ④ よりきめ細やかな画像を実現するためにグラフィックスオブジェクト上で情報の補完をする処理

問 3 下線部③に関連して、IEEE 方式の倍精度浮動小数点数で 2.5 をあらわした 2 進数表示として適切なものを ① ~ ④ から一つ選べ。

- ① 01000000001000000000000000000000
- ② 11000000001000000000000000000000
- ③ 0100000000000000000000000100000000
- ④ 1100000000000000000000000100000000

### III 参考文献

- 1 Wikipedia
- 2 コミックマーケット公式サイト (<http://www.comiket.co.jp/>)
- 3 東京国際展示場 (<http://www.bigsight.jp/>)
- 4 ゲームプログラマになる前に覚えておきたい技術
- 5 DeepTech([http://news.cnet.com/8301-30685\\_3-20000277-264.html](http://news.cnet.com/8301-30685_3-20000277-264.html))
- 6 ZENet Japan  
<http://builder.japan.zdnet.com/news/story/0,3800079086,20410291,00.htm>

### IV SpecialThanks

Moneto 氏 (<http://moneto.blogspot.com/>)