

Q1 (10 点)

ID: fpoint/text01/page01/011

0 以上の整数を表す 16 進数 0x 1f を 2 進数に変換した時の値を選択肢 a～d の中から 1 つ選びなさい。

(a)

0b 0111 0111

(b)

0b 1111 0011

(c)

0b 0001 1111

(d)

0b 1001 1001

Q1 (10 点)

ID: fpoint/text01/page01/011

正解 (c)

【出題意図】

0 以上の 10 進数の整数を 2、16 進数と相互変換することができるかどうかを確かめる問題である。

【重要事項】

極めて基本的な事項なので省略する

【解説】

極めて基本的な事項なので省略する

Q2 (10 点)

ID: fpoint/text01/page02/001

10 進数 -19 を 2 の補数を使って 8 ビットの 2 進数に変換した時の値を選択肢 a~d の中から 1 つ選びなさい。

(a)

0b 1110 1101

(b)

0b 0100 1100

(c)

0b 1110 0100

(d)

0b 1010 0111

Q2 (10 点)

ID: fpoint/text01/page02/001

正解 (a)

【出題意図】

10 進数の負を含む整数を 2 の補数を使って 2、16 進数と相互変換することができるかどうかを確かめる問題である。

【重要事項】

変換方法

・ 10 進数 → 2 進数

ステップ 1. 0 以上の整数の場合はそのまま 2 進数を求めて終わり

ステップ 2. マイナス符号を取り除いた絶対値の 2 進数を求める

ステップ 3. 長さが指定されたビット数未満の場合は先頭に 0 を追加して指定されたビット数にする

ステップ 4. 各桁の 0 と 1 をひっくり返す

ステップ 5. 1 を足す

・ 2 進数 → 16 進数

ステップ 1. 4 ビットごとに区切る。4 ビットに満たない所は 0 で埋める

ステップ 2. 0～F までの 16 進数を割り当てる

・ 2、16 進数 → 10 進数

ステップ 1. 16 進数の場合は一度 2 進数に戻す

ステップ 2. 先頭ビットが 0 の場合は 0 以上の値なのでそのまま 10 進数を求めて終わり

ステップ 3. 1 を引く

ステップ 4. 各桁の 0 と 1 をひっくり返す

ステップ 5. 10 進数を求める

ステップ 6. 先頭にマイナス符号を付ける

【解説】

重要事項より求まる。

Q3 (10 点)

ID: fpoint/text01/page02/011

16 進数の 0x E (2 の補数形式、4 ビット) を 10 進数の整数に変換した時の値を選択肢 a~d の中から 1 つ選びなさい。

(a)

3

(b)

-14

(c)

-5

(d)

-2

Q3 (10 点)

ID: fpoint/text01/page02/011

正解 (d)

【出題意図】

10 進数の負を含む整数を 2 の補数を使って 2、16 進数と相互変換することができるかどうかを確かめる問題である。

【重要事項】

変換方法

・ 10 進数 → 2 進数

ステップ 1. 0 以上の整数の場合はそのまま 2 進数を求めて終わり

ステップ 2. マイナス符号を取り除いた絶対値の 2 進数を求める

ステップ 3. 長さが指定されたビット数未満の場合は先頭に 0 を追加して指定されたビット数にする

ステップ 4. 各桁の 0 と 1 をひっくり返す

ステップ 5. 1 を足す

・ 2 進数 → 16 進数

ステップ 1. 4 ビットごとに区切る。4 ビットに満たない所は 0 で埋める

ステップ 2. 0～F までの 16 進数を割り当てる

・ 2、16 進数 → 10 進数

ステップ 1. 16 進数の場合は一度 2 進数に戻す

ステップ 2. 先頭ビットが 0 の場合は 0 以上の値なのでそのまま 10 進数を求めて終わり

ステップ 3. 1 を引く

ステップ 4. 各桁の 0 と 1 をひっくり返す

ステップ 5. 10 進数を求める

ステップ 6. 先頭にマイナス符号を付ける

【解説】

重要事項より求まる。

Q4 (10 点)

ID: fpoint/text01/page03/011

0 以上かつ 1 より小さい小数を表す 2 進数 0b 11001 (有効桁数 $n = 5$ ビット) を 10 進数の小数に変換した時の値を選択肢 a~d の中から 1 つ 選びなさい。

(a)

$$0.78125 \left(= \frac{1}{2} + \frac{1}{4} + \frac{1}{32} \right)$$

(b)

$$0.375 \left(= \frac{1}{4} + \frac{1}{8} \right)$$

(c)

$$0.28125 \left(= \frac{1}{4} + \frac{1}{32} \right)$$

(d)

$$0.5 \left(= \frac{1}{2} \right)$$

Q4 (10 点)

ID: fpoint/text01/page03/011

正解 (a)

【出題意図】

0 以上かつ 1 より小さい 10 進数の小数を 2、16 進数と相互変換することができるかどうかを確かめる問題である。

【重要事項】

極めて基本的な事項なので省略する

【解説】

極めて基本的な事項なので省略する

Q5 (10 点)

ID: fpoint/text02/page01/002

10 進数 $17.5625 (=17 + 1/2 + 1/16)$ を固定小数点数形式を使って 16 進数に変換した時の値を選択肢 a~d の中から 1 つ選びなさい。小数点以下の有効桁数は 4 ビットとする。

(a)

0x 11.D

(b)

0x 11.9

(c)

0x 11.F

(d)

0x 11.5

Q5 (10 点)

ID: fpoint/text02/page01/002

正解 (b)

【出題意図】

10 進数の小数を固定小数点数形式の 2 進数または 16 進数と相互変換することができるかどうかを確かめる問題である。

【重要事項】

変換方法

・ 10 進数 → 2 進数

ステップ 1. 整数部と小数部で別々に 2 進数を求める

ステップ 2. 整数部の 2 進数と小数部の 2 進数を足し合わせると固定小数点数形式の 2 進数になる

・ 2 進数 → 16 進数

ステップ 1. 整数部と小数部で別々に 4 bit ごとに区切る。4bit に満たない場合は 0 で埋める

ステップ 2. 0～F までの 16 進数を割り当てる

・ 2、16 進数 → 10 進数

ステップ 1. 16 進数の場合は一度 2 進数に戻す

ステップ 2. 整数部と小数部で別々に 10 進数を求める

ステップ 3. 整数部の 10 進数と小数部の 10 進数を足し合わせると求める 10 進数になる

【解説】

整数部が $17 \rightarrow 0x11$ 、小数部が $0.5625 \rightarrow 0x0.9$ となることより求まる。

Q6 (10 点)

ID: fpoint/text02/page01/011

固定小数点数形式で表される 2 進数 0b 101.0001 を 10 進数に変換した時の値を選択肢 a～d の中から 1 つ選びなさい。

(a)

$$5.75 = \left(5 + \frac{1}{2} + \frac{1}{4}\right)$$

(b)

$$5.0625 = \left(5 + \frac{1}{16}\right)$$

(c)

$$5.625 = \left(5 + \frac{1}{2} + \frac{1}{8}\right)$$

(d)

$$5.0$$

Q6 (10 点)

ID: fpoint/text02/page01/011

正解 (b)

【出題意図】

10 進数の小数を固定小数点数形式の 2 進数または 16 進数と相互変換することができるかどうかを確かめる問題である。

【重要事項】

変換方法

・ 10 進数 → 2 進数

ステップ 1. 整数部と小数部で別々に 2 進数を求める

ステップ 2. 整数部の 2 進数と小数部の 2 進数を足し合わせると固定小数点形式の 2 進数になる

・ 2 進数 → 16 進数

ステップ 1. 整数部と小数部で別々に 4 bit ごとに区切る。4bit に満たない場合は 0 で埋める

ステップ 2. 0～F までの 16 進数を割り当てる

・ 2、16 進数 → 10 進数

ステップ 1. 16 進数の場合は一度 2 進数に戻す

ステップ 2. 整数部と小数部で別々に 10 進数を求める

ステップ 3. 整数部の 10 進数と小数部の 10 進数を足し合わせると求める 10 進数になる

【解説】

重要事項より求まる。

Q7 (10 点)

ID: fpoint/text03/page01/011

10 進数 1.5 を IEEE754(単精度) 形式を使って 2 進数に変換した時の値を選択肢 a～dの中から 1 つ選びなさい。

(ヒント) $0.5 = 1/2$

(a)

0b 0 00001110
0011000000000000000000

(b)

0b 0 11101100
1100100000000000000000

(c)

0b 1 10100110
1001111000000000000000

(d)

0b 0 01111111
1000000000000000000000

Q7 (10 点)

ID: fpoint/text03/page01/011

正解 (d)

【出題意図】

10 進数の 0 以上の小数を浮動小数点数形式 (IEEE754、単精度) の 2、16 進数と相互変換できるかどうかを確かめる問題である。

【重要事項】

変換方法

・ 10 進数 → 2 進数

ステップ 1. 「符号部」を求める。10 進数の小数が 0 以上なら 0、負なら 1

ステップ 2. 10 進数の小数の「絶対値」の固定小数点数形式の 2 進数を求める

ステップ 3. ステップ 2 で求めた 2 進数の小数点を左右に移動して整数部を 1 桁の 1 だけにする。移動した回数を E とする (右方向がマイナス)。小数部分を M とする。

ステップ 4. 「指数部」(単精度の場合は 8 ビット) を求める。単精度の場合は、ステップ 3 で求めた E に 127 を足して 8 ビットの 2 進数に変換する

ステップ 5. 「仮数部」(単精度の場合は 23 ビット) を求める。単精度の場合は、ステップ 3 で求めた M が 23 ビットよりも長い場合は 24 ビット目が 0 なら 24 ビット目を切り捨て、1 なら 24 ビット目を切り上げる (※)

23 ビットより短い場合は後ろを 0 で埋める ※この方式の事を「最近接偶数丸め」言う。単純に 24bit 目以降を切り捨てる場合もある

ステップ 6. 符号部、指数部、仮数部を繋げ、単精度の場合は 32 bit の 2 進数にする

【解説】

重要事項より求まる。

Q8 (10 点)

ID: fpoint/text03/page02/011

IEEE754(単精度) 形式において、符号部が 0 か 1、指数部が全て 1、仮数部が 0 以外の時の値を選択肢 a~d の中から 1 つ選びなさい。

(a)

+Infinity

(b)

NaN

(c)

-0

(d)

-Infinity

Q8 (10 点)

ID: fpoint/text03/page02/011

正解 (b)

【出題意図】

IEEE754 形式における特殊な値を求めることができるかどうかを確認める問題である。

【重要事項】

値	符号部	指数部	仮数部
+0	0	全て 0	全て 0
-0	1	全て 0	全て 0
+Infinity	0	全て 1	全て 0
-Infinity	1	全て 1	全て 0
NaN	0 または 1	全て 1	0 以外

【解説】

重要事項より求まる。

Q9 (10 点)

ID: fpoint/text03/page03/010

C 言語で変数 a が単精度浮動小数点数型である時、丸め誤差の影響を小さくするために有効な方法を選択肢 a～d の中から 1 つ選びなさい。

(a)

コンピュータのメモリを増やす

(b)

半精度にする

(c)

倍精度にする

(d)

変数の名前を変える

Q9 (10 点)

ID: fpoint/text03/page03/010

正解 (c)

【出題意図】

浮動小数点数形式で生じる誤差について理解しているかどうかを確認める問題である。

【重要事項】

・ 丸め誤差

小数を含む実数を固定小数点数形式や浮動小数点数形式で 2 進数化すると生じる誤差。

精度を上げたり、計算方法を工夫することである程度防ぐことが出来る。

・ 桁落ち

近い値の小数同士で引き算をすると仮数部の有効桁数が減ることで生じる誤差。

精度を上げたり、計算方法を工夫することである程度防ぐことが出来る。

【解説】

精度を上げる (ビット数を増やす) ことで丸め誤差の影響を小さくすることができる。

Q10 (10 点)

ID: fpoint/text03/page03/011

整数 12 を float 型変数に代入した時の丸め誤差を選択肢 a～d の中から 1 つ選びなさい。

(a)

0.0

(b)

12.0

(c)

0.5

(d)

-1.2

Q10 (10 点)

ID: fpoint/text03/page03/011

正解 (a)

【出題意図】

浮動小数点数形式で生じる誤差について理解しているかどうかを確認める問題である。

【重要事項】

・ 丸め誤差

小数を含む実数を固定小数点数形式や浮動小数点数形式で 2 進数化すると生じる誤差。

精度を上げたり、計算方法を工夫することである程度防ぐことが出来る。

・ 桁落ち

近い値の小数同士で引き算をすると仮数部の有効桁数が減ることで生じる誤差。

精度を上げたり、計算方法を工夫することである程度防ぐことが出来る。

【解説】

整数は丸め誤差が生じない。