【１】浮動小数点数の表現形式　　　　　　　　　　　　　　　　　　　　　　…テキストP.（　　　）

　整数データは固定小数点数の形式で表現するのに対して、[①　　　　　　　]は浮動小数点数の形式で表現する。なお、形式には**単精度**（32ビット）と**倍精度**（64ビット）がある。

例　浮動小数点数の表現形式（単精度；32ビット）

|  |  |  |
| --- | --- | --- |
| ０ | ０００ ００１０ | ００１０ ０１００ ００００ ００００ ００００ ００００ |

▲

符号(１ビット)

指数部(7ビット)

仮数部(24ビット)

【２】浮動小数点数の正規化と表現　　　　　　　　　　　　　　　　　　　　…テキストP.（　　　）

　浮動小数点数は、数値Ｘを「Ｘ＝Ｍ×ＢE」のように指数の形で表現する方法で、Ｍを[①　　　　]、Ｂを[②　　　　　　　　　　]、Ｅを[③　　　　　]と呼ぶ。ここで、仮数部Ｍを「0.1≦Ｍ＜１」に収める。このことを**正規化**という。

例１　正規化（0.1≦Ｍ＜１）

基本 ： （１２３）10　 ⇒　（0.123）10×103

◆２進浮動小数点数　（１１０１）2　 ⇒　[④　　　　　　　　　　　　　　]

◆16進浮動小数点数　（６Ｂ）16　 ⇒　[⑤　　　　　　　　　　　　　]

例２　浮動小数点数の表現

◆２進浮動小数点数　（１１０１）2　⇒　（０．１１０１）2×２4

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | …⑥ |

◆16進浮動小数点数　（６Ｂ）16　⇒　（０．６Ｂ）16×１６2

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | …⑦ |

例３　次の形式の浮動小数点数で10進数の（１）100、（２）0.375を表現せよ。

|  |  |  |
| --- | --- | --- |
| Ｓ | Ｅ | Ｍ |

　　　1ビット　　　　7ビット　　　▲　　　　　　　　　24ビット

　　　Ｓ：仮数部の符号（正は０、負は１）

　　　Ｅ：指数部（２を基数とし、負数は2の補数で表現）

　　　Ｍ：仮数部（2進数の絶対値で表現）

１．（100）10を正規化した浮動小数点数で表現する。

　(1)10進数から2進数へ変換

　　　（100）10 ＝（0110 0100）2

　(2)（0110 0100）2を指数表現

　　　（0110 0100）2 ＝（0110 0100）2×２0

　(3)仮数部の最上位が1となるように正規化

　　　（0110 0100）2×２0 ＝（0.110 0100）2×２7

　(4)浮動小数点数で表現

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | …⑧ |

２．（0.375）10を正規化した浮動小数点数で表現する。

　(1)10進数から2進数へ変換

　　　（0.375）10 ＝（0.011）2

　(2)（0.011）2を指数表現

　　　（0.011）2 ＝（0.011）2×２0

　(3)仮数部の最上位が1となるように正規化

　　　（0.011）2×２0 ＝（0.11）2×２-1

　(4)浮動小数点数で表現

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | …➈ |

　　　※指数が―1の負数であるため、2の補数で表現！

【３】演算精度と誤差　　　　　　　　　　　　　　　　　　　　　　　　　　…テキストP.（　　　）

１．指数部で発生する誤差

　浮動小数点数で、**非常に絶対値の大きな値同士の乗算**を行ったりすると、指数部で表現できる最大値を越えてしまうことがある。これを[①　　　　　　　　　　]という。

　逆に、**絶対値の小さな値（非常に0に近い値）同士の乗算**を行ったりすると指数部で表現できる最小値を表現できない。これを[②　　　　　　　　　]という。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 負の  Over Flow | 表示できる指数 | 負の  Under Flow | 正の  Under Flow | 表示できる指数 | 正の  Over Flow |
|  | 負　←←←　　０　　→→→　正 | | | |  |

２．仮数部で発生する誤差

（１）[③　　　　　　　]

　例えば、1/3＝0.3333…のように無限小数となり、コンピュータに記憶可能な桁数を超える計算においては、ある桁の次の桁以降に対して、切り捨て、四捨五入、切り上げなどの端数処理が行われる。端数処理を行うことを「丸める」といい、このとき**真値との間に生ずる誤差**を丸め誤差と呼ぶ。

（２）[④　　　　　　　]

　ある程度の値で収束が確認できたところで、処理を打ち切ることで生じる誤差であり、このとき**真値との間に生じる誤差**を打切り誤差と呼ぶ。

（３）[⑤　　　　　　　]

　絶対値の差が非常の大きい2つの数値の加減算を行った場合、**絶対値の小さいほうの値が無視**され、演算結果に反映されないために発生する誤差である。

例　仮数部のけた数が6桁の浮動小数点数で、0.123456×106＋0.123454×100を行う場合。

0.123456 ×106

+） 0.123454 ×100　　⇒

0.123456　　　 ×106

+） 0.000000123454 ×106

0.123456123454 ×106

　　　　無視される

　情報落ちを減らす工夫としては、情報落ちは絶対値の小さいほうの指数を、大きいほうに合わせる過程で発生する。そこで、**絶対値の小さい順に数値を並び替え、絶対値の差のない数値同士の演算を先に行い**、ある程度の大きさになってから絶対値の大きな値を加減すると誤差を減らせる。

（４）[⑥　　　　　　　]

値がほぼ等しくかつ**丸め誤差**を持つ数値同士の減算を行った結果、**有効数字が減少すること**を桁落ちと呼ぶ。

例　桁落ち

　　　 0.556×107

－） 0.552×107

結果 0.004×107　→　0.4**00** × 105

この計算結果では、正規化の結果有効桁数が3桁から1桁に減少する。しかし実は「0.4**00**」の **00**は強制的に0を埋められた結果であり、本当はこの部分は00以外の何か有効な値があったはず！

≪範例1≫

浮動小数点演算において，絶対値の大きな数と絶対値の小さな数の加減算を行ったとき，絶対値の小さな数の有効けたの一部又は全部が結果に反映されないことを何というか。

ア　打切り誤差 　　イ　けた落ち 　　ウ　情報落ち 　　エ　絶対誤差

≪解答≫　ウ

ア　打切り誤差とは、無限和や無限小数の計算を途中で打ち切ることによって発生する誤差です。

イ　けた落ちとは、浮動小数点演算の絶対値のほぼ等しい数値の減算において、上位の有効数字が失われることによって生じる誤差です。

エ　絶対誤差とは、計算値などから真の値を代数的に引いた結果です。

３．絶対誤差と相対誤差

絶対誤差は、真値との差の絶対値であり誤差自身の大きさである。 相対誤差は真値に対する**割合で示した誤差**であり、誤差が測定値に占める割合。

例　以下の絶対誤差と相対誤差を求めよ。

Ａ：真値が１００ｍで誤差が０．５ｍ

Ｂ：真値が５００ｋｍで誤差が１ｋｍ

解答

絶対誤差について

Ａ：[⑦　　　　　　]

Ｂ：[⑧　　　　　　]

相対誤差について

Ａ：０．５／１００＝[⑨　　　　　]

Ｂ：　１　／５００＝[⑩　　　　　]

≪範例2≫

三つの実数Ｘ～Ｚとそれぞれの近似値が次の場合、相対誤差の小さい順に並べたものはどれか。

|  |  |  |
| --- | --- | --- |
|  | 真の値 | 近似値 |
| Ｘ | 1.02 | 1 |
| Ｙ | 1.97 | 2 |
| Ｚ | 5.05 | 5 |

ア　Ｘ，Ｙ，Ｚ 　　　イ　Ｙ，Ｚ，Ｘ 　ウ　Ｚ，Ｘ，Ｙ 　　　エ　Ｚ，Ｙ，Ｘ

≪解答≫　エ

相対誤差は、真値に対する割合で示した誤差で、次の式で求めます。

相対誤差＝｜真値－近似値｜／｜真値｜

この式に従って、Ｘ、Ｙ、Ｚの相対誤差を求めると次のようになります。

Ｘの相対誤差＝｜1.02－1｜／｜1.02｜≒0.02

Ｙの相対誤差＝｜1.97－2｜／｜1.97｜≒0.015

Ｚの相対誤差＝｜5.05－5｜／｜5.05｜≒0.01