令和３年度「専修学校による地域産業中核的人材養成事業」

専門学校と高等学校の有機的連携プログラムの開発・実証

工業系分野における高専連携の５年一貫教育プログラム開発・実証

SE・IT分野

［骨子案］

学校法人小山学園 **専門学校東京工科自動車大学校**

はじめに

本事業は、文部科学省から委託された「専修学校による地域産業中核的人材養成事業」の内のひとつである「専門学校と高等学校、教育委員会等の行政及び企業が協働で、高・専一貫の教育プログラムを開発するモデルを構築する」事業の成果報告書である。

この事業は、中等教育（高等学校）の段階から、キャリア意識を高め、専門知識を涵養できるようなアプローチをすることが、専門学校における学びの質を高めることに繋がるとともに、高校生にとっては職業意識を醸成することによって、将来の就業に適した専門知識や技術を習得する進路を導き出すことが可能となる。

これは、高卒就職者の約４割が離職するという統計やその後の離転職においてキャリアアップを図れず、非正規雇用のままが続く、というケースの改善にもつながる。

本校は、1969 年開校の自動車整備士養成の工業系の専門学校として、約４万人以上の卒業生を自動車業界へ輩出している。また、同法人内に「建築」、「インテリア」、「情報・IT・Web」、「AI・データサイエンス」、「バイオテクノロジー」、「環境」の分野（学科）を有する専門学校 東京テクニカルカレッジ」(東中野)を設置おり、日本の基幹産業を網羅している。また、114 社（2022 年 1 月現在）の企業が加盟している「後援会」組織が、就職や教育課程編成委員会等の活動で協力をしてくれている。

今回の事業は、この幅広い職業分野をカバーしている専門学校と高校、さらに企業群と教育連携することによって、実社会に即した職業教育と高専一貫の教育プログラムを開発することで、高校生段階から職業観を図れることと高度な専門知識や技術の習得へ繋がり、確かな就職活動、その後のキャリア形成に役立つものと確信している。

また、Society5.0 や SDGs、DX 等の社会的インフラ、教育課程においては、GIGA スクール構想をはじめ学習指導要領の改定等、初等中等教育から高等教育までの教科や学習の仕方が大きく変わろうとしている。

本事業は、「高専一貫」として、高校の３年間と専門学校の２年から５年間という、長い期間の教程をカバーするため、刻々と変化する時代のニーズを反映しなければならない。

今回は初年度ということから、高校、専門学校、企業、行政等の各視点から、この事業の必要性と方向性を各種調査やヒアリング等を行い、その結果を成果報告書として編集している。

最後に、今回の報告書を多くの高校、専門学校、企業がこれからの高校、専門学校のあり方の道標としていただければ幸いです。

事業責任者
学校法人小山学園 専門学校 東京工科自動車大学校
校長 佐々木 章

目 次

コンピュータシステム１_２コマ参考箇所

【コンピュータシステム１_２コマ参考箇所】

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

参考２コマ

◆数値の表し方

コンピュータの中では2進数を使って数値が表されます。64ビットコンピュータの演算装置では、最大64けたの2進数を表すことができます。実際に扱う数値はこれより小さい場合、大きい場合があります。64ビット、ということは下の図で表したように64個のけたが並んでいて、それぞれに数を入れて同時に計算できるということになります。

そろばんや電卓の表示を考えるとわかりやすいですね。電卓だとおおよそ10けたくらいしかないので、すごく大きく感じますが、電卓やそろばんでは1つのけたに0～9の数が入るのに対して、コンピュータのそろばんには０か１どちらかしか入らないのです。

10進数とは・・・　0～9の、10個の数字を使って数を表す方法
２進数とは・・・　0～1の、2個の数字をつかって数を表す方法

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

■10進数では・・・

0から始めて、数を1ずつ増やしていくと1、2、3、4、5、6、7、8、9までは1けたで表せます。

さらに増やすともう1けたで表せず、10と書きます。これを日本語では「じゅう」と発音します。10進数では9を越える数の場合、けたを増やしてあらわします。

■2進数では・・・

0から始めて、数を1ずつ増やしていくと1までは1けたで表せます。さらに増やすともう1けたで表せず、10と書きます。これは日本語では「いち-ぜろ」と発音します。

なぜか。10進数だから10を「じゅう」と読んだんです。2進数の場合、読み方はないので「いち-ぜろ」と読むしかないのです。2進数では１を越える数の場合、けたを増やしてあらわします。

■ならば・・・

2以上の自然数（正の整数）なら何進数でも作れます。コンピュータは0と1しか使えないデジタル回路でできているので、2進数を使います。

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

■16進数は・・・

人間の指が10本あるから10進数が数えやすかったそうなんですが、2進数は人間にとってはけた数が多すぎて苦痛です。そこで2進数の4けたをひとまとめに１つの数字で表す16進数が、コンピュータの世界では良く使われます。あくまで人間が2進数を扱いやすくするためのもので、16進数で計算する要素は、実はコンピュータにはありません。

64ビットを16進数なら16文字で表すことができます。ただそれだけだけど、とても大切です。

・16進数とは・・・0～9、およびA～Fまでの16個の数字を使って数を表す方法ということになります。9より大きな数字はないので、A～Fを数字として使ってしまうのです。

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

◆10進数

10進数（Decimal Number）は、0から9までの10個の数字を使って数を表現します。

数は、0、1、2、3、4、5、6、7、8、9と順に増え、次に位が増えて10になります。このようにして、10進数は、1、10、100、1000、10000…と位が繰り上がります。

たとえば10進数で２９７６という数は、どのように表されているのでしょうか。

数値の表現	2	9	7	6
位（くらい）	1000の位	100の位	10の位	1の位
乗数での位の表現	10^3	10^2	10^1	10^0

これは、数式で以下のように表すことができます。

$2\times10^3 + 9\times10^2 + 7\times10^1 + 6\times10^0 = 2\times1000 + 9\times100 + 7\times10 + 6\times1 = 2976$

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

◆2進数

2進数（Binary Number）は、数字0,1の2個の数字を使って数を表現します。
数は、0、1と順に増え、次に位が増えて10になります。「じゅう」と読まないように。
このようにして、2進数は、１、２、４，８…と位が繰り上がります。

例えば2進数で1101という数は、どのように表されているでしょうか。

数値の表現	1	1	0	1
位（くらい）	8の位	4の位	2の位	1の位
乗数での位の表現	2^3	2^2	2^1	2^0

これは、数式で以下のように表すことができます。

$1\times2^3 + 1\times2^2 + 0\times2^1 + 1\times2^0 = 1\times8 + 1\times4 + 0\times2 + 1\times1 = 13_{(10)}$

※数値の右下に $_{(10)}$ と書いてあれば10進数表記であることを示します。

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

10進数では1の位、10の位、100の位…を私たちが知っているように、2進数では右から1の位、2の位、4の位、8の位…を知っていれば2進数を10進数にすることは比較的簡単にできます。

逆に、10進数から2進数へ変換するには、10進数を2で割って、その商をさらに2で割る、またその商を2で割って…と、余りを出しながら商が0になるまで繰り返します。
そして最後の余りを先頭に下から順に並べます。
例えば10進数で19という数は、以下のように計算します。

19) 2 = 9 … 1	19を2で割る → 9 あまり 1
9) 2 = 4 … 1	9を2で割る → 4 あまり 1
4) 2 = 2 … 0	4を2で割る → 2 あまり 0
2) 2 = 1 … 0	2を2で割る → 1 あまり 0
1) 2 = 0 … 1	1を2で割る → 0 あまり 1
計算結果は 10011 $_{(2)}$	上の筆算の「あまり」を下から順に書く。 1÷2まで書かないと間違えるので注意

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

◆16進数

16進数（Hexadecimal Number）は、0から9までの数字とAからFまでのアルファベットを使って数を表現します。

数は、0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、Fと順に増え、次に位が増えて10になります。

Aは10進数で10、Bは10進数で11、Cは10進数で12、Dは10進数で13、Eは10進数で14、Fは10進数で15を表す数字になります。

このようにして、16進数は1、16、256、4096…と位が繰り上がります。

例えば4E5Fという16進数は、どのように表されているでしょうか。

数値の表現	4	E	5	F
位（くらい）	4096の位	256の位	16の位	1の位
乗数での位の表現	16^3	16^2	16^1	16^0

これを、数式で次のように表すことができます。

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

$4\times16^3 + E\times16^2 + 5\times16^1 + F\times16^0$

$= 4\times4096 + E\ (14)\times256 + 5\times16 + F\ (15)\times1 = 20063_{(10)}$

※数値の右下に $_{(10)}$ と書いてあれば10進数表記であることを示します。

16進数を10進数に手で計算するときは、以下のように筆算します。

4 × 4096 =	16384	16^3のけたは	4096の桁だから
E × 256 =	3584	16^2のけたは	256の桁だから
5 × 16 =	80	16^1のけたは	16の桁だから
F × 1 =	＋） 15	16^0のけたは	1の桁だから
	20063	これを合計する	

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

10進数から16進数へ変換するには、10進数を16で割って、その商をさらに16で割る、またその商を16で割って…と、余りを出しながら商が0になるまで繰り返します。
そして最後の余りを先頭に下から順に並べます。
例えば10進数で1000という数は、以下のように計算することができます。

```
1000 ) 16  =  62 ･･･  8   1000を16で割る → 62 あまり  8
  62 ) 16  =   3 ･･･ 14     62を16で割る →  3 あまり 14
   3 ) 16  =   0 ･･･  3      3を16で割る →  0 あまり  3
```

計算結果は　3E8(16)、筆算の余りを下から順に書く。ただし14(10)はE (16)

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

◆10進数 ⇔ 2進数 ⇔ 16進数の対応
※桁位置がわかりやすいように、２進数では４桁ずつ揃えてあらわしています。

10進数	2進数	16進数
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F
16	0001 0000	10
17	0001 0001	11
18	0001 0010	12
19	0001 0011	13
20	0001 0100	14

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

◆8進数

これまでの授業で16進数が出てきました。16進数は直接コンピュータで処理を行うものではないけれど、2進数を人間にも読みやすくするため4桁を1桁の数字で表現したものでした。

最近はあまり使われなくなってきましたが、同じ用途で8進数があります。16進数が2進数4けたずつ区切るのに対して、8進数は3けたずつ区切ります。

現在でもC言語では8進数が使えます。またUNIXやLINUXではコマンドを操作するとき8進数が必要になります。

8進数…　0～7の8つの数字を使って数を表す方法

コンピュータシステム１_４コマ参考箇所

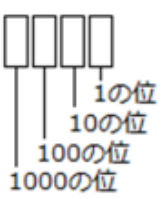
【コンピュータシステム１_４コマ参考箇所】

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

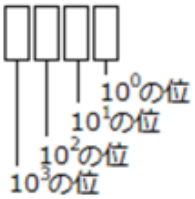
参考４コマ

◆整数各けたの重み

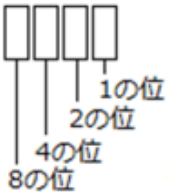
10進整数では、それぞれの桁を示すのに1の位（くらい）、10の位…などと表現し、一けた左に移動するたびに10倍になります（図1）。これをべき乗で表すと100の位、101の位、102の位になり、一けた左に移動するたびに乗数が1増えます（図2）。



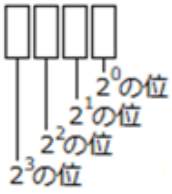
（図1）



（図2）



（図3）



（図4）

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

これが2進数だと、桁を示すのに1の位(これは変わらない)、2の位…と表現し、一けた左に移動するたびに2倍になります(図3)。べき乗なら20の位、21の位、22の位になり、一けた左に移動するたびに乗数が1増えます。

10進数で1234という数を表した場合は、

$(1234)_{10} = 1\times10^3 + 2\times10^2 + 3\times10^1 + 4\times10^0$

2進数で1101という数を表した場合は、

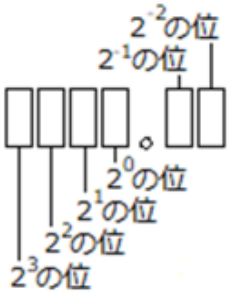
$(1101)_2 = 1\times2^3 + 1\times2^2 + 0\times2^1 + 1\times2^0$

となりますが、数の表し方として各けたに掛けた数を「重み」といいます。

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

◆小数各けたの重み

10進数でも2進数でも整数の場合一番小さいけたの重みは１でした。
10進数なら左に一けたずれるごとに10倍、2進数なら2倍になりました。
小数の場合は、10進数なら右に一けたずれるごとに1/10になるので、2進数なら1/2になることがわかります。大切なのは、小数点のすぐ左が何進数でも「重み1」であることです。



工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

表1．10進数、2進数の各けたの重み

10進数	1000の位 10^3	100の位 10^2	10の位 10^1	1の位 10^0	小数点	0.1の位 10^{-1}	0.01の位 10^{-2}
2進数	8の位 2^3	4の位 2^2	2の位 2^1	1の位 2^0		0.5の位 2^{-1}	0.25の位 2^{-2}

10進数で12.34という数を表した場合は、

$(12.34)_{10} = 1\times10^{1} + 2\times10^{0} + 3\times10^{-1} + 4\times10^{-2}$

2進数で101.01という数を表した場合は、

$(101.01)_{2} = 1\times2^{2} + 0\times2^{1} + 1\times2^{0} + 0\times2^{-1} + 1\times2^{-2}$

となります。大切なのは小数点のすぐ左のけたが1の位（10^0 または 2^0）であることです。

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

◆2進小数を10進数に変換する

2進小数を10進小数に変換するには、上式を使います。

$(101.01)_2 = 1\times2^2 + 0\times2^1 + 1\times2^0 + 0\times2^{-1} + 1\times2^{-2}$
$= 1\times4 + 0\times2 + 1\times1 + 0\times0.5 + 1\times0.25$
$= 4 + 1 + 0.25 = (5.25)_{10}$

$(1110.101)_2 = 1\times2^3 + 1\times2^2 + 1\times2^1 + 0\times2^0 + 1\times2^{-1} + 0\times2^{-2} + 1\times2^{-3}$
$= 1\times8 + 1\times4 + 1\times2 + 0\times1 + 1\times0.5 + 0\times0.25 + 1\times0.125$
$= 8 + 4 + 2 + 0.5 + 0.125 = (14.625)_{10}$

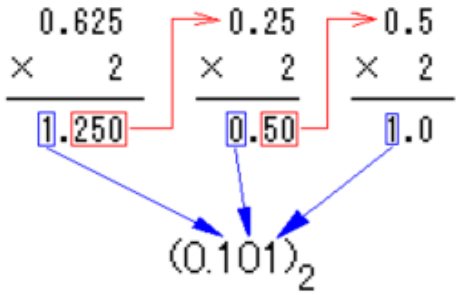
工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

◆10進小数を2進数に変換する

10進小数を 2 倍し、結果のうち小数部分だけを 2 倍していき、結果の小数部分が 0 になるまで続けます。そして整数部分を並べて答えを求めます。ただし、計算が無限に続くことがあります。

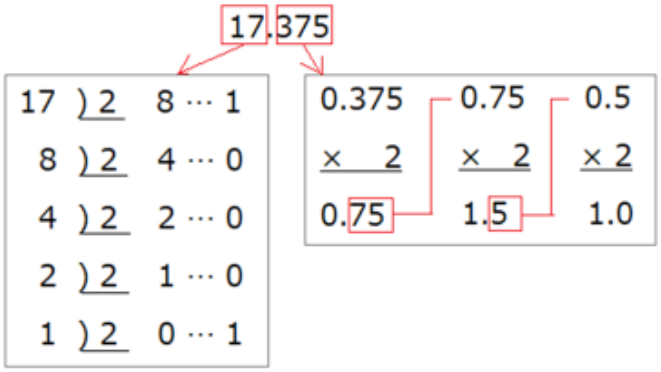
※10進小数では普通の小数でも、2進数に変換すると無限小数になることがあります。

小数部だけの例　$(0.625)_{10} = (0.101)_2$



工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

整数部と小数部がある例（17.375）$_{10}$ ＝ （1 0001．011）$_{2}$
下図のように整数部と小数部を分けて、別々に筆算します。



工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

◆2進小数と16進小数の相互変換

16進数は2進数4けたを1文字であらわすことができます。なので2進数を4けたずつ区切れば簡単に相互変換できます。注意すべきなのは、小数点を中心に必ず4けたずつ区切らなければならないことです。

2進小数から16進小数への変換例（0011 1011.0101 1）$_{2}$ ＝ （3B.58）$_{16}$

1. 小数点を中心に、右に4けた、左に4けたずつ区切ります。
2. 左端は16進数で3です。
3. 次の4けたは1011なのでB。
4. 小数点の右4けたは0101なので5。
5. 右端に1が残りますが、これは4けたにすると1000なので8。

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

16進小数から2進小数への変換例（5C.46）$_{16}$ ＝（0101 1100.0100 011）$_{2}$

1. 5を2進数に変換すると101。
2. Cは1100。このとき4けたずつすきまを空けるとわかりやすい。
3. 小数点と、その右に0100。
4. 6は0100。そのまま4けた書いても合ってるけど、普通終わりの0は書かない。

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

◆10進小数と16進小数の変換

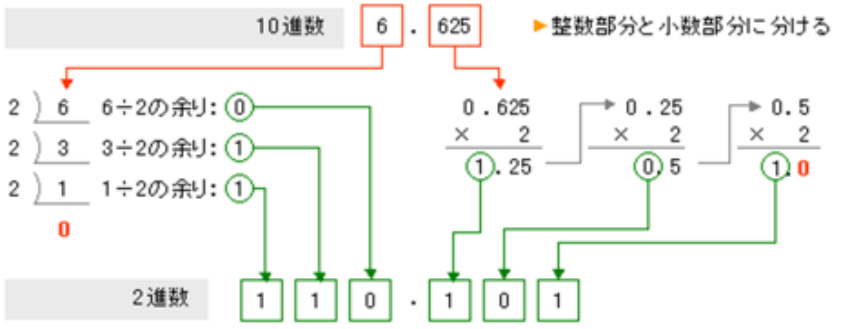
理屈さえわかれば10進小数と16進小数の変換もできます。
直接でなく、一旦2進小数を介した変換も、楽に計算ができます。

（0.625）$_{10}$ → （0.A）$_{16}$
0.625を16倍すれば10.0となり、これは16進1けた（A）$_{16}$であらわせます。

$$
\begin{aligned}
(0.A)_{16} &= 10 \times 16^{-1} \\
&= 10 \times 0.0625 \\
&= (0.625)_{10}
\end{aligned}
$$

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

◆小数点数の基数変換



▶整数部分は、商が0になるまで2で割り続ける。計算過程で発生する余りが2進数の各桁の値となる。

▶小数部分は、2を掛け続ける。掛けたものの整数部分を2進数の各桁の値とし、この計算結果での少数部分が0となれば終了。

コンピュータシステム１_７コマ参考箇所

【コンピュータシステム１_７コマ参考箇所】

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

参考７コマ

補数による負の数の表現

2進数で負の数を表すには、次の2通りの方法があります。

１．数の絶対値に符号を追加する方法

２．２の補数による方法

コンピュータ内部では、正数と同じように演算ができる「２の補数」により負の数を表すのが一般的です。

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

◆10進３けたで正負の数を表すと

10進数3けたを表示するはかりをイメージしましょう。３けたの数字と、これをアナログ表示する目盛りがついています（図1）。このはかりで測れるのは0gから999gまでの1000通りの数です。999gを1g超えると0gになってしまい、これをオーバフローといいます。

図1．10進3けたで表せる正数

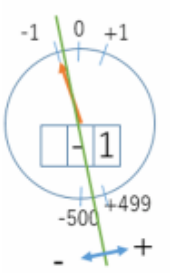


図2．10進３けたで10の補数を使って表せる正負の数

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

こんな使い方は普通しませんが、はかりの天秤を手で引っ張るとどうなりますか？仮に1gの力で引っ張れば999gと表示されるはずです。この引っ張る行為を「マイナスの重さをはかった」ことにすればどうでしょう。すなわち999g ⇒ -1gと読み替えるのです（図2）。表せる範囲を下表に示します。どちらも1000通りの数になります。

	正数として測れる範囲	正負の数として測れる範囲
測れる最大値	999	+499
中間の値	500	0
測れる最小値	0	-500

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

このはかりで-20gをはかると、どう表示されるでしょうか。次の計算から980と表示されることがわかります。
赤字のＩは「オーバフロー」が起きたと考えるためです。

```
  1000
-   20
  980
```

またはオーバフローを考えずに、与えられた３けたの範囲で考えると、次のように計算できます。

```
  999        979
-  20      +   1
 [979]      [980]
```

この場合の979を「9の補数」、1をたした980を「10の補数」といいます。

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

◆8ビット2進数で「2の補数」を使って正負の数を表すと

左の例と同じように２進数についても考えることができます。１０進数の例では１０進３けたとしましたが、ここでは２進数８ビットで考えることにします。けた数さえ考えていれば何ビットあっても同様に考えることができます。

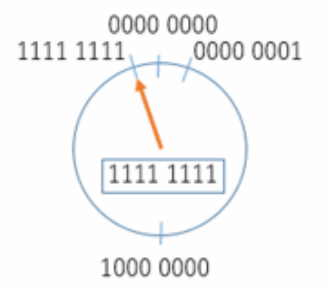


図3．8ビット2進数で表せる正数

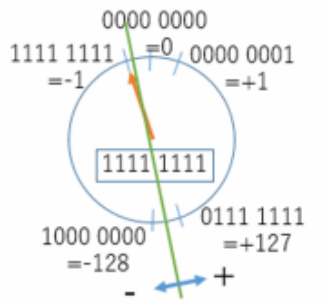


図4．8ビット2進数で２の補数を使って表せる正負の数

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

	正数として測れる範囲		正負の数として測れる範囲	
測れる最大値	1111 1111	255	0111 1111	+127
中間の値	1000 0000	128	0000 0000	0
測れる最小値	0000 0000	0	1000 0000	-128

-100を８ビット２の補数で表すには、先ず１の補数を求め、これに１をたして２の補数にします。

```
 1111 1111          1001 1011
-0110 0100        +         1
 1001 1011          1001 1100
```

左が１の補数、右が2の補数です。
左の計算を見ると繰り下がりがなく、各けたの「0」と「1」が入れ替わっているのがわかります。
すなわち1⇒0、0→1と、数字を反転させればよいのです。

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

◆足し算引き算では2の補数でも正数と同じように計算できる

ただしオーバフローしないこと。オーバフローとは数値が表せる範囲を越えることをいいます。

◆１けたシフトすれば2倍、または1/2倍になる

10進数990を10倍、または1/10にするには、１けた左、または右に移動すればよいことがわかります。

990 → 9900 10倍　　990 → 99 1/10

2進数でも同様に2倍、または1/2にするには、1けた左または右に移動すればよいのです。

0001 1000 → 0011 0000 2倍　　0001 1000 → 0000 1100 1/2

※ただしオーバフローすれば結果はおかしくなります。

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

2の補数の場合は1/2にするときだけ操作が異なります。最上位けたが符号を表すため、負数のとき0を入れるとおかしくなるためです。

	すべて正数の場合（unsigned）	2の補数の場合(正負の数：signed)
×2	捨てる ← □□□□ □□□□ ← 0	□□□□ □□□□ ← 0 ↓符号はそのまま ※ □↓□□□ □□□ 0　符号の1つ右を捨てる
1/2	0 → □□□□ □□□□ → 捨てる	□　□□□ □□□□ ↓ □→□□□ □□□□ → 捨てる

※情報処理技術者試験で使用される架空のコンピュータCOMETⅡでの表現に合わせました。

コンピュータシステム１_８コマ参考箇所

【コンピュータシステム１_８コマ参考箇所】

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

参考８コマ

◆論理シフト

論理シフトはビットパターンを2進数としてではなく、あくまでビットの並びとして操作することを念頭に命名されていますが、正の整数を２^{n}倍または1/2^{n}にすることにも使われます。

右論理シフト：ビットパターンを右にシフトし、空いた桁を0で埋めます。正の整数を1/2にします。
左論理シフト：ビットパターンを左にシフトし、空いた桁を0で埋めます。正の整数を２倍します。

右論理シフト	左論理シフト
0 → □□□□ □□□□ → 捨てる	捨てる ← □□□□ □□□□ ← 0

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

右論理シフトで余ったビットは捨てられます。捨てられたビットが1の場合、正の整数を1/2にした結果、小数第1位を切り捨てたことになります。（アンダーフローとは呼ばないことに注意）
例：0000 0111を1ビット右論理シフトした結果は0000 0011

左論理シフトで余ったビットは捨てられます。捨てられたビットが1の場合、正の整数を2倍した結果、演算結果がオーバフローした（けたあふれした）ことになります。
例：1000 0000を1ビット左論理シフトした結果は0000 0000

◆算術シフト

ビットパターンを２の補数で表される正負の整数としてシフト操作を行います。その結果正負の整数を２^{n}倍または2^{-n}（1/2^{n}）にすることができます。

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

右算術シフト：ビットパターンを右にシフトし、空いた桁には最上位けたと同じ「0」または「1」が入ります。

左算術シフト：ビットパターンを左にシフトし、空いた桁を0で埋めます。符号ビットは変えず、その一つ右のビットが捨てられます。

右算術シフト	左算術シフト
□□□□ □□□□ ↓ □→□□□ □□□□ → 捨てる	□□□□ □□□□ ← 0 ↓ 符号ビットの１つ右を捨てる （符号ビットは変えない）

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

右算術シフトで余ったビットは捨てられます。捨てられたビットは1の場合正負の整数を＜元の数を越えない整数＞に切り捨てたことになります。

例：1111 1101を1ビット右算術シフトした結果は1111 1110

左算術シフトを行った結果、表現できる数値範囲を越得た場合、符号は変わりませんが数値はおかしくなります。

例：1000 0000を1ビット左算術シフトしても結果は1000 0000

-128_{10} → -128_{10}

0111 1111を1ビット左算術シフトすると結果は0111 1110

$+127_{10}$ → $+126_{10}$

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

◆指数表記

私たちはとても大きな、あるいは0に近い数を表すとき、指数表記を使います。
例えば真空中の光の速さを秒速30万kmとすると、これを

3.00×10^8 m/s

と表すことができます。これは3の後ろに0が8個つくことを表しています。
指数表記には3つの要素があります。

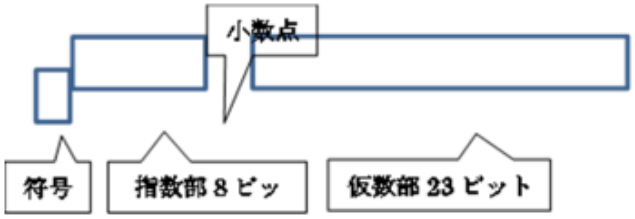
符号：＋
仮数：3.00
指数：8

この表現方法をコンピュータ内部でも採用したのが浮動小数点形式です。

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

◆単精度浮動小数点形式

指数表記を符号1ビット、仮数部23ビット、指数部8ビットの合計32ビットの2進数で表したものを単精度浮動小数点形式といいます。



例：+10．25を単精度浮動小数点形式で表す。

1. 符号は＋なので0
2. 10．25を固定小数点2進数に変換すると、1010．01
3. これを1．xxxの形になるように小数点を３けた左に移動すると、1．0100１となる。

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

4．整数部の1を除いた小数部が仮数となる。

0100 1000 0000 0000 0000 000

5．移動したけた数3にオフセット127を足した値が指数部となる。（オフセットバイナリ形式）

1000 0010

+10．25を単精度浮動小数点形式で表すと、

0100 0001 0010 0100 0000 0000 0000 0000 となる。

他の数値例：

数値	符号	指数部	仮数部
+1．0	0	0111 1111	0000 0000 0000 0000 0000 000
+2．0	0	1000 0000	0000 0000 0000 0000 0000 000
+10．0	0	1000 0010	0100 0000 0000 0000 0000 000
−10．0	1	1000 0010	0100 0000 0000 0000 0000 000
0	0	0000 0000	0000 0000 0000 0000 0000 000

※仮数部は整数部分が常に1である「1．xxxx・・・」の形となるように調整され、これを正規化といいます。
ただし数値「0」だけは仮数部の整数部分を1にできません。0のときだけ例外的に全て0で表します。

コンピュータシステム１_１０コマ参考箇所

【コンピュータシステム１_１０コマ参考箇所】

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

参考10コマ

◆10進数データ表現

これまで10進数を2進数に変換する方法、および2進数での加減算について学んできました。

一方で、なるべく10進数の形のまま処理を行う方法があり、次のような利点があります。

1．10進数と2進数の変換が要らない。

2．処理に必要な10進けた数を任意にとることができる。

◆BCDコード

最初に登場したマイクロコンピュータは4ビットであり、その意図する用途はプログラム電卓でした。

4ビットあれば10進数１けたを表現することができ、プログラムによって四則演算や、その他の関数演算を行うことができます。また電卓以外にも、計算はしないが１０進数を使用する機器は数多くあります。

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

このように10進数1桁を4ビット2進数で表し、これをけた数分並べて表現する方法を2進化10進コード、またはBCD(Binary Coded Decimal)コードと呼びます。表１はBCDコード１桁を表していますが、１０１０以上の数をBCDでは使用しません。

10進1桁	2進数4ビット
0	００００
1	０００１
2	００１０
3	００１１
4	０１００
5	０１０１
6	０１１０
7	０１１１
8	１０００
9	１００１

表1．２進化１０進コード（BCDコード）

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

◆2進化10進（BCD）演算機構

キーボードなどから入力された数値は、ゾーン10進数として記録されています。ゾーン10進数は、そのままの形で演算することはできず、プログラムによってパック10進数に変換したのち演算できるようになります。

CPUには「2進化10進演算」のための演算機構があり、パック10進数を処理することができます。

そして処理が行われたパック10進数をディスプレイやプリンタに出力するためには、プログラムでゾーン10進数に戻します。

BCD演算機構では、2進演算に加えて4ビットごとに桁上がりが行われ、「１００１」に１を加えたとき、さらに「０１１０」が加算されるようになっています。

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

◆ゾーン１０進数

１バイトを上位４ビットの「ゾーン部」と、下位４ビットの「数値部」に分け、１バイトで10進数１けたを表します。ゾーン10進数のことを「アンパック10進数」と呼ぶ場合もあります。

ゾーン10進数でｎけたの数値を表す場合ｈ、ｎバイト必要になります。

最下位バイトのゾーン部には正負を表す「符号」が入りますが、それ以外のけたでは文字コードと同じになります。

このため、数値と文字列への変換が容易にできます。

１．＋１２３４の場合

４けたあるので４バイト必要です。正の数の場合、全てのゾーン部は同じになります。

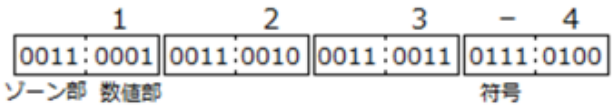
また、それぞれのけたは文字コード“１’、‘２’、‘３’、‘４’と同じです。

※JISコードの場合

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

２．－１２３４の場合

４けたあるので４バイト必要です。負数の場合は最下位けたのゾーン部が0011から0111に変わります。



工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

◆パック10進数

1バイトに２けたを記憶するため、バイト数を減らすことができます。また演算時には数けた分を同時に計算できるため処理にかかる時間も（わずかですが）短くなります。

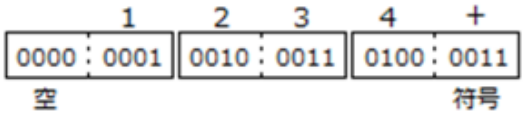
パック10進で数値を表すのに必要なバイト数は、ｎけたの場合 [ｎ / 2] ＋ 1 バイト必要です。

ただし[ ]内は切り捨てです。

１．＋１２３４の場合

最終けたの下位４ビットに符号を表す「0011」が入り、その右側に4ビットを10進数１けたが入ります。

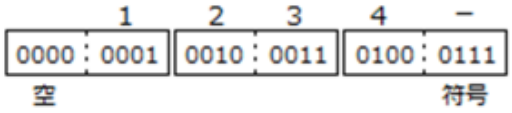
いちばん左側の上位4ビットは、この場合むだになります。



工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

２．－１２３４の場合

最終けたの下位4ビットに符号を表す「0111」が入る以外は、＋１２３４と同じになります。



コンピュータシステム１_１１コマ参考箇所

【コンピュータシステム１_１１コマ参考箇所】

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

参考11コマ

◆論理式

コンピュータの演算機構は「算術論理演算装置：ALU（Aritmetic　Logic Unit）」と呼ばれ、数値計算だけでなく、論理演算も行えるようになっています。論理演算を行うことで、ビット毎に細かい操作を加えることができます。
論理演算が行えるから条件分岐などの処理が行える、とも言えます。

またコンピュータ全体のほとんどの部分が論理式を扱うための回路、論理回路でできています。四則演算を行う算術演算機構や、CPU内にあってデータを一時保持する「レジスタ」なども論理回路でできています。

論理回路：論理式の演算を行うための回路
論　　理：物事を筋道立てて考えるときの考え方
論 理 式 ：ＡＮＤ、ＯＲ、ＮＯＴ演算を使って組み立てられた式。

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

1．数理計算における関数をイメージしよう
通常の一次関数は次のような形になります。

$Y = A \cdot X + B$

一方論理式では、次の形が一般形（積和形式）になります。

$Y = A \cdot X + B$　　なあんだ、同じじゃないか。

ちなみにAND（論理積）は記号「・」または無印、 OR（論理和）は「＋」であらわすので、ここまでは論理式も同じ書き方になります。

2．論理式では、変数や定数は０か１どちらかの値しかとらない。
変数Aが１のときYも１になるのであれば、

$Y = \overline{A}$

ところで変数Aが０のときYが１になるのであれば、次のように書きます。

$Y = A$

これはAが0のとき、その反対である1となることをAの上の棒（Bar）で示しています。

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

3．変数が右辺に複数ある場合

例えば次のように書きます。

$Y = \overline{A}\,\overline{B} + A\overline{B} + AB$

上式は、次のことを表しています。

Yは、　A＝０、B＝１であるか、

A＝１、B＝０であるか、

A＝１、B＝１のとき、１になる。

AはA＝１のとき、$\overline{A}$は$\overline{A}$＝０のときと読みます。

それでは論理演算の演算子を見ていきましょう。

どのような結果になるかを表す方法に「真理値表」「ベン図」があり、この2つで演算を理解します。

1．真理値表：入力変数が取りうる全ての組み合わせを表にして、結果を並べたもの。

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

2．長方形の中にいくつかの丸を描き、１となる組み合わせを図示するもの。
なお数値０は真（True）、数値１は偽（False）と呼ぶ場合があります。
「論理記号」は論理回路図で使用する記号です。

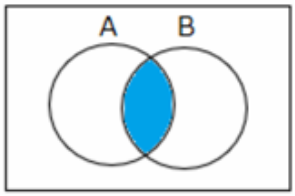
◆論理積（AND）

最低2個ある入力値が全て１のとき、結果が１となる演算です。
それ以外の入力組み合わせは、全て０となります。

ANDの真理値表

A	B	A ・ B
0	0	0
0	1	0
1	0	0
1	1	1

ANDのベン図



論理記号

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

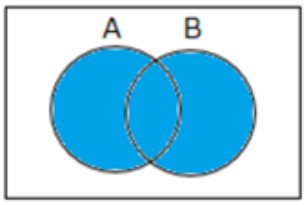
◆論理和（OR）

最低2個ある入力値が全て0のとき、結果が0となる演算です。
それ以外の入力組み合わせは、全て1となります。

ORの真理値表

A	B	A ＋ B
0	0	0
0	1	1
1	0	1
1	1	1

ORのベン図



論理記号

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

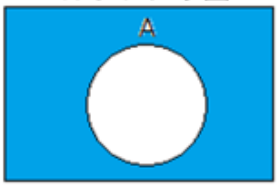
◆否定（NOT）

入力は一つしかありません。入力が0なら結果は1です。
論理記号の末尾についている○が、「否定」を表しています。

ＮＯＴの真理値表

A	$\overline{A}$
0	1
1	0

ＮＯＴのベン図



論理記号

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

◆論理演算の用例（マスク）

良く使われる論理演算（ビット演算）に「マスク」があります。
「マスク」は何ビットもある2進データのうち、ＡＮＤ演算を使って必要な部分だけを拾い出すときに使います。マスクを使って不要な部分を（隠す）イメージです。

1．16ビットデータのうち下位8ビットに注目する

```
  1000 0101 1010 1100 ← もともとあるデータ
ＡＮＤ 0000 0000 1111 1111 ← マスク
  0000 0000 1010 1100 ← 取り出されたデータ
```

不要な部分を全て0にするため、もともとあるデータと共にＡＮＤ演算を行うデータをマスクと呼びます。

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

2．16ビット2進数の符号に注目する

```
  0111 1010 0010 1111 ← 正負を判定したいデータ
ＡＮＤ 1000 0000 0000 0000 ← マスク
  0000 0000 0000 0000 ← 先頭ビットが０なので正
```

数値の符号が正か負かだけを知りたい場合には、先頭ビットだけが1であるマスクを使用します。

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

◆論理演算の用例２（文字コード変換）

文字コード表を見ると、数字やアルファベットが一貫性をもって並んでいることがわかります。
ここではアルファベット大文字と小文字変換に論理演算を応用してみます。

1．大文字を小文字に変換
大文字と0x20をOR演算することで、小文字に変換できます。

```
    0100 0001 ← 大文字の'A'
OR  0010 0000 ← 0x20
    0110 0001 ← 小文字の'ａ'
```

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

2．小文字を大文字に変換
小文字と0xDFをAND演算することで、大文字に変換できます。

```
    0110 0001 ← 小文字の'ａ'
ＡＮＤ  1101 1111 ← 0xDF
    0100 0001 ← 大文字の'A'
```

3．大文字小文字を逆にする
文字と0x20とを排他的論理和をとると、大文字は小文字に、小文字は大文字に変換できます。

```
    0110 0001 ← 小文字の'ａ'          0100 0001 ← 大文字の'Ａ'
ＥＯＲ  0010 0000 ← 0x20        ＥＯＲ  0010 0000 ← 0x20
    0100 0001                         0110 0001 ← 小文字の'ａ'
```

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

◆ド･モルガンの定理

論理式を扱う上で重要な定理がド･モルガンの定理です。

$$\overline{A \cdot B} = \overline{A} + \overline{B}$$

$$\overline{A + B} = \overline{A} \cdot \overline{B}$$

この定理はベン図を使って説明することができます。

注意：Ｃ言語ではここで使用している演算を「ビット演算」といいます。
「論理演算」はｉｆ文などの条件式に使用する演算子を意味するので、注意が必要です。

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

論理式の変換法則（ブール代数）

論理式は X = A +B という形で表記され、X は出力を、A と B は入力を表します。	
交換の法則	$A+B=B+A$　　$A \cdot B=B \cdot A$
結合の法則	$A+(B+C)=(A+B)+C$　　$A \cdot (B \cdot C)=(A \cdot B) \cdot C$
恒等の法則	$A+1=1$　$A \cdot 1=A$　　$A+0=A$　　$A \cdot 0=0$
同一の法則	$A+A=A$　　$A \cdot A=A$
補元の法則	$A+\overline{A}=1$　　$A \cdot \overline{A}=0$
復元の法則	$\overline{\overline{A}}=A$
ド・モルガンの法則	$\overline{A+B}=\overline{A} \cdot \overline{B}$　　$\overline{A \cdot B}=\overline{A}+\overline{B}$
分配の法則	$A \cdot (B+C)=A \cdot B+AC$
吸収の法則	$A+A \cdot B=A$　　$A \cdot (A+B)=A$

コンピュータシステム１_１２コマ参考箇所

【コンピュータシステム１_１２コマ参考箇所】

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

参考12コマ

◆データ構造

これまで学習してきたデータを複数集めることで、意味のあるひとまとまりの情報とすることができます。このときのデータの置き方をデータ構造といい、対象とするデータによっていくつかの構造がよく用いられます。
データ構造を考える際には、メモリ上の配置も考える必要があります。

◆スタック

データの出し入れを考えたとき、最後に入れたデータが最初に取り出される（LIFO：Last In First Out）性質を持つデータ構造で、割り込み制御や関数呼び出しを行う際有用なことから、ほとんどすべてのコンピュータはメモリ上でスタック構造を扱うためのレジスタ「SP：スタックポインタ」を持っています。

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

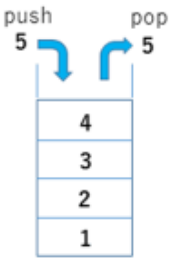


図1．スタック構造

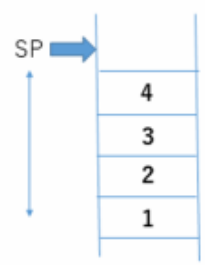


図2．メモリ上のスタック構造

図1ではスタックにデータ1，2，3，4が入っていて、これから5を入れる状態を示しています。データをスタックに入れることをpush、取り出すことをpopといいます。最後に5をpushした場合、すぐ後にpopされるのは5になります。
図2は、このスタックをメモリ上で構成した場合を表します。はしごのようなものはメモリの一部だと考えてください。スタックに1，2，3，4が入っていて次にpushする番地をSP:Stack Pointerが示しています。

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

データ5がpushされると、SPがさす番地に5が入り、SPが一つ前の番地に移動します。popはその逆で、SPが一つ後の番地に移動してから、SPがさす番地の内容が取り出されます。

図3．pushの流れ図

図4．popの流れ図

図3にデータを入力する流れ図を示します。
図4にデータを取り出す流れ図を示します。pushしたデータより多くのデータをpopしようとする「スタックアンダーフロー」に注意する必要があります。

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

◆キュー

データの出し入れを考えたとき、最初に入れられたデータが一番最初に出てくる（FIFO：First In First Out）性質を持つデータ構造で、プリンターへ送るデータなど、データを一時保存する際によく使われます。

図5．キュー構造

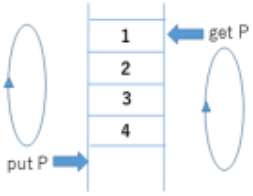


図6．メモリ上のキュー構造

図5において、キューに入力される（enqueue）データは図の左側から入り、図の右側から（dequeue）取り出されます。キューの大きさ（保存されているデータ数）は増減します。

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

図6ではキューをメモリ上で実現する方法を示します。データをenqueueするためのポインタputP（プットポインタ）とdequeueするためのポインタgetP（ゲットポインタ）の2つのポインタが必要です。

putPは次にenqueueする番地を示していて、キューに許された範囲を越えると、キューの先頭番地に戻り、先頭から入力します。メモリ上で輪のように構成されることから、リングバッファと呼ばれることがあります。

getPは次にdequeueする番地を示していて、putP同様に範囲を越えそうになると先頭番地に戻ります。

なおgetP、putPはプログラム上で作成する必要があります。

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

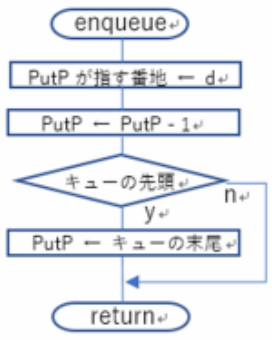


図7．データを入力する流れ図

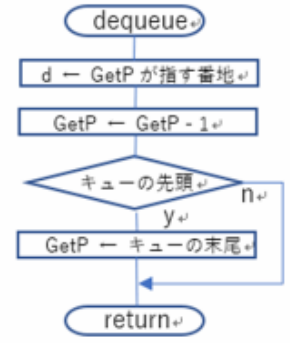


図8．データを取り出す流れ図

図7にキューへデータを入力する流れ図を示します。データがキューバッファからあふれなよう制御が必要です。

図8にキューからデータを取り出す流れ図を示します。入力された以上にデータを取り出さないよう制御が必要です。

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

◆配列

スタックやキューはデータを出し入れするときの順番が決まっていましたが、番号を指定することでどのデータでも出し入れできるのが配列です。

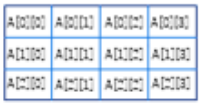


図9．2次元配列

図10．メモリ上の2次元配列

プログラムを作成するうえで、配列は非常によく使われます。図9はC言語で2次元配列を扱うときのイメージです。Aという配列の要素を2つの添え字で指定しています。配列は1次元から原則的に何次元でも作ることができます。

図10では、この2次元配列をメモリ上で構成する方法を示しています。メモリ上の配列をアクセスするために、CPU内にハードウェアとして持っているインデックスレジスタ（IX：IndeX Register）が使われ、配列要素の位置を指定するために、インデックスレジスタ上でアドレスの算出が行われます。

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

□木

木をさかさまにしたような形で、階層構造を表現するのに適したデータ構造です。図11において、一番上の始点にあたる要素を「根：ルート」、下方の端にあたる要素を「葉：リーフ」、その他の要素を「節：ノード」と呼び、根や節の下には0から複数の節または葉が連なっています。

いろいろな木構造を構成できますが、代表的なものに「二分探索木」があります。キーを探索するのに、ルートから始まって今見ているノードよりもキーが大きければ右、小さければ左に進みます。図12においてキー5を探索し、見つからなければ二分探索木に追加するとします。ルートの8から探索を始め、5ではないので4の右下に追加します。

木構造の場合、次の要素を示すために要素内に「ポインタ」が置かれます。そのためメモリ上のデータの配置はどのようになっていても構いません。要素を追加する場合でも、移動することなくメモリ上に追記できます。要素を削除する場合もポインタの変更だけで済むため、いらなくなった要素はそのままメモリ上に残されます。

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

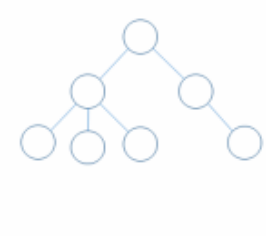


図11．一般的な木構造

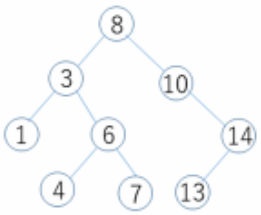


図12．2分探索木の例

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

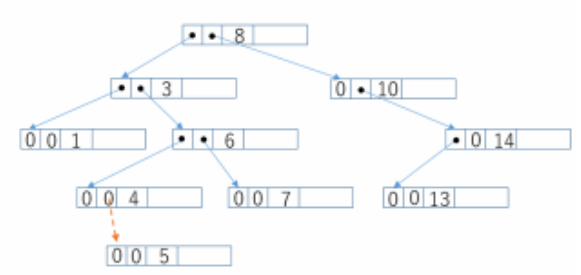


図13．2分探索木への要素追加

図13のように、二分探索木の一つの要素は「右ポインタ」「左ポインタ」「キー」「その他のデータ」からなっています。要素が葉であり、その下に続く要素がない場合はポインタの値をゼロとします。

キー5の要素を追加するには、その上の節の右ポインタを0から、追加する要素のアドレスに変更します。

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

◆リスト

木構造のように上から1対多ではなく、1対1で1本に連なっているデータ構造をリストといいます、図14に一般的なリスト構造を示します。各要素はセルと呼び、先頭のセルを指しているポインタが一つあります。

このリストには核セルにキー値が保存されていて、キーの昇順にセルが並んでいるとします。ここにキー8を持つセルを挿入することを考えます。キーの値から真ん中のセルと終端のセルの間に挿入する必要があることがわかります。

このときは図15のように真ん中のセルの次ポインタと、挿入セルの次ポインタを変更することで挿入を行います。

図のようにキーとなる数値で順序を指定する場合がありますが、一致する文字列や、リスト上何番目かでセルを指定する場合があります。

工業系分野における高専連携の５年一貫教育プログラム開発・実証（ＳＥ・ＩＴ分野）

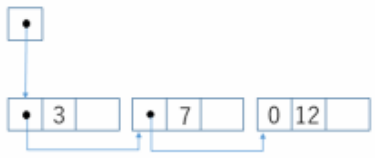


図14．リスト構造（単方向）

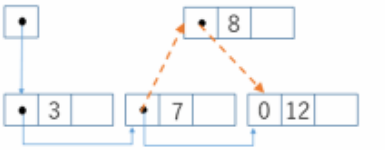


図15．リストにセルを挿入する

木構造の場合と同じく挿入の場合でもポインタの変更だけで、要素の移動はおきません。削除の場合もポインタの修正だけで削除が行われるため、要らなくなった要素はメモリ上にたまっていきます。そのため挿入削除を繰り返したあと、時々要らなくなったセルを削除する操作「ガベージコレクション」を行う必要があります。

MEMO

本「ＳＥ・ＩＴ分野骨子案」は、文部科学省の教育政策推進事業委託費による委託事業として、≪学校法人小山学園 専門学校東京工科自動車大学校≫が実施した令和３年度「専修学校による地域産業中核的人材養成事業」の成果をとりまとめたものです。

令和３年度「専修学校による地域産業中核的人材養成事業」
専門学校と高等学校の有機的連携プログラムの開発・実証

工業系分野における高専連携の５年一貫教育プログラム開発・実証　ＳＥ・ＩＴ分野骨子案

令和　４年　３月発行

発行所・連絡先

学校法人小山学園 専門学校東京工科自動車大学校
〒164-0001 東京都中野区中野 6-21-16
TEL 03-3360-8824　FAX 03-3360-8805
https://car.ttc.ac.jp/