基礎ソフト 実験レポート2

C過程 S-15組 1TE20137W 2022/10/19

柳 鷹

問4

(1)

求めるプログラムは以下である。

|  |
| --- |
| \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*\*  \*\* アドレスレジスタa1 で指定されるアドレスから始まる連続する2nバイトのメモリの内容をアドレスレジスタa2で指定されるアドレスから始まる領域へと転送するサブルーチンの作成  \*\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  .section .text  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*\* メインルーチン  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  start:  jsr COPY /\* サブルーチンへ \*/  stop #0x2700 /\* プログラム終了 \*/  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*\* データを転送するサブルーチン  \*\* a1:読み込みデータのアドレス  \*\* a2:書き込みデータのアドレス  \*\* d0:書き込み回数  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  COPY:  movem.l %d0-%d7/%a0-%a6,-(%sp) /\* レジスタ退避 \*/  lea.l READ, %a1 /\* 読み込みデータREADの先頭アドレスをa1レジスタへ \*/  lea.l WRITE, %a2 /\* 書き込み先WRITEの先頭アドレスをa2レジスタへ \*/  moveq.l #0, %d0 /\* 書き込み回数初期化 \*/  LOOP:  move.b (%a1)+, (%a2)+ /\* READからWIRTEへ \*/  addq.l #1, %d0 /\* 書き込み回数 +1 \*/  cmp.l #LENGTH, %d0 /\* 書き込み回数 < LENGTH ならば \*/  bne LOOP /\* LOOPへ \*/  movem.l (%sp)+,%d0-%d7/%a0-%a6 /\* レジスタ復帰 \*/  rts /\* サブルーチンから復帰 \*/  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*\* データエリア  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  .equ LENGTH,20  .section .data  READ:  .ascii "TAROU " /\* NAME \*/  WRITE:  .ds.b 20 /\* 書き込みデータ出力先 \*/  .end |

(2)

プログラムの目的：

アドレスレジスタa1 で指定されるアドレスから始まる連続する2nバイトのメモリの内容をアドレスレジスタa2で指定されるアドレスから始まる領域へと転送するサブルーチンの作成

レジスタ用途：

a1:読み込みデータのアドレス

a2:書き込みデータのアドレス

d0:書き込み回数

メモリ内のデータ配置：

0x000400 番地から 0x00042d まではテキスト領域で、0x000430 番地から 0がデータ領域（0x000430+データサイズまでが読み込みデータ、それ以降書き込み領域）である。

(3)

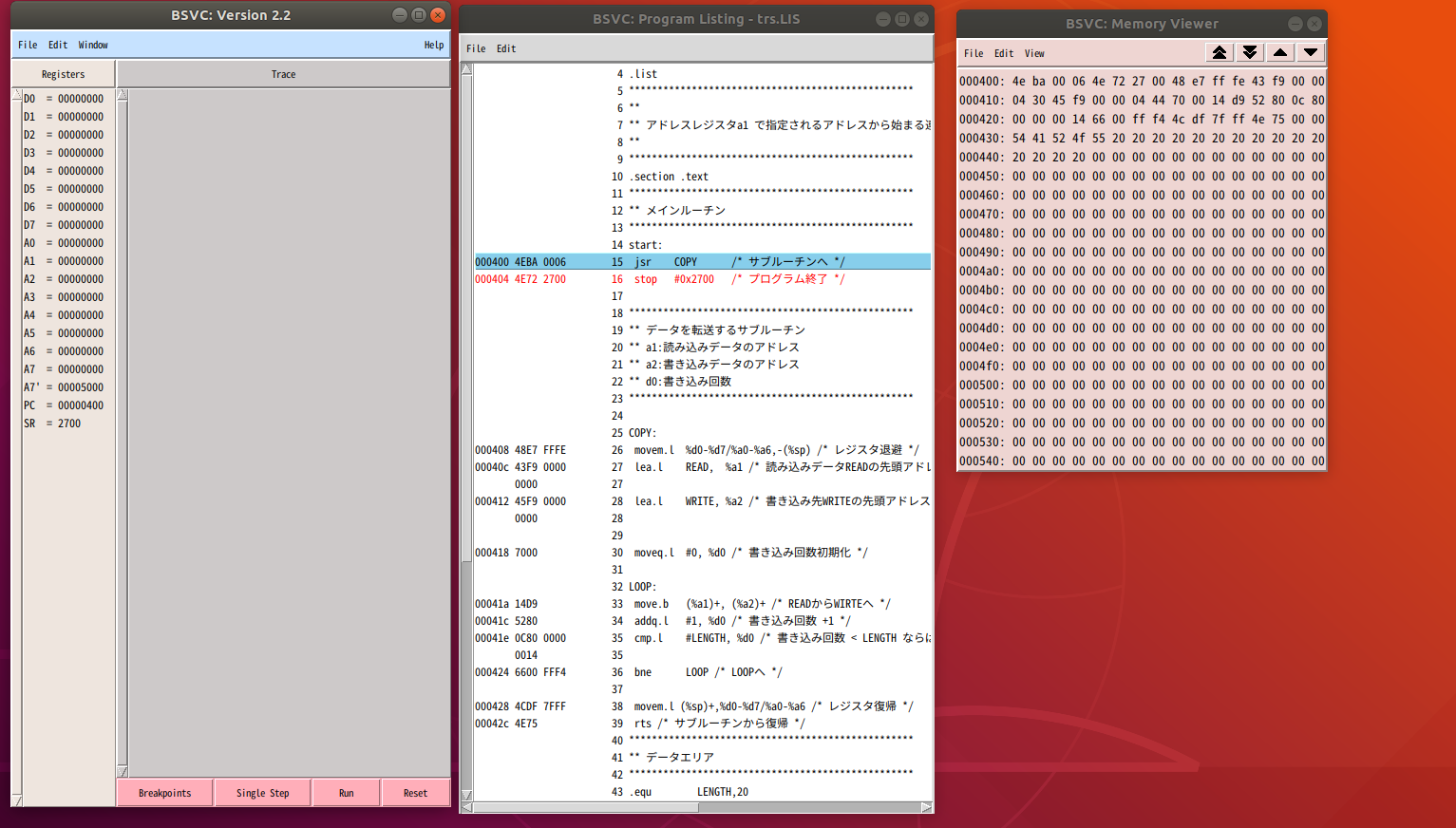
・読み込みデータ（転送元）を「.section .data　READ:」以下に記述する

・読み込みデータのサイズ（何文字か）をシンボルであるLENGTHに入れる

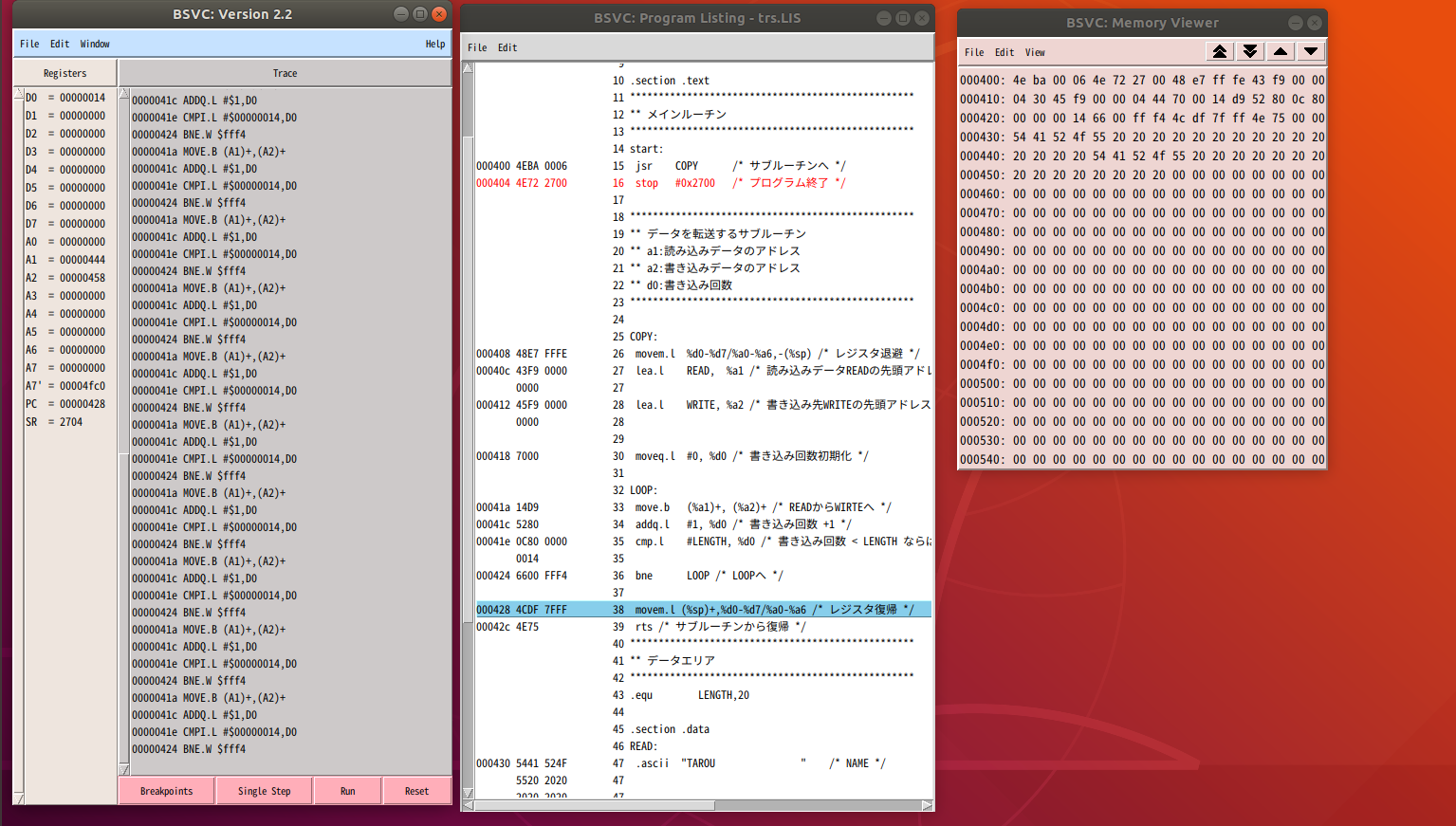
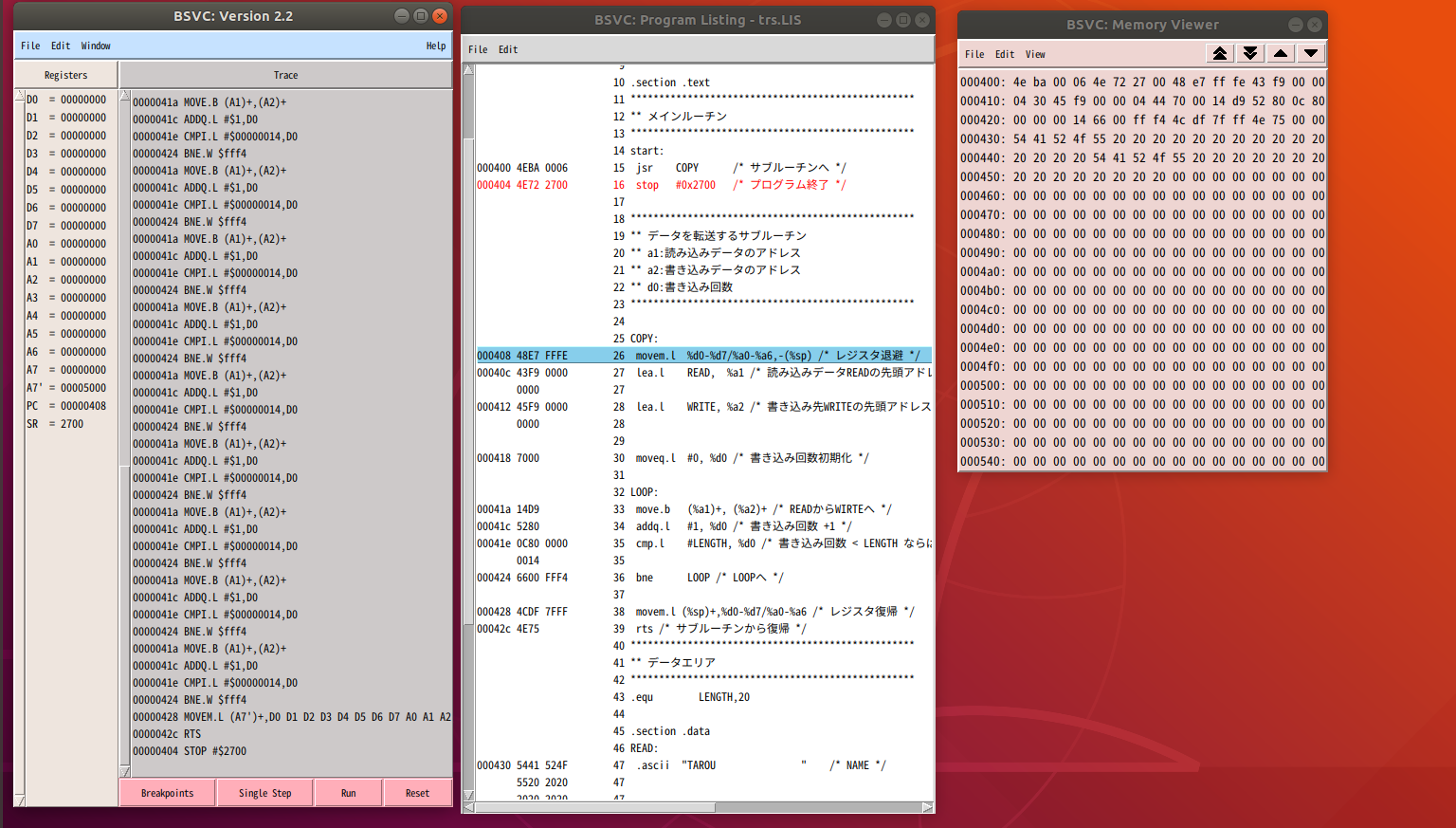
（ASCIIコードに変換して2バイトを1としている）

(4)

プログラムの実行結果をメインルーチン実行前、サブルーチンレジスタ復帰直前、メインルーチン実行後のメモリ・レジスタの様子を比較しながら示す。

「メインルーチン実行前」

「サブルーチンレジスタ直前」

「メインルーチン実行後」

まず、メインルーチン実行前とサブルーチンレジスタ復帰前を比較すると、青枠で囲まれたメモリ（0x000444番地～0x000457番地）に0x000430番地～0x000443番地のデータが転送されているのが確認できる。

また、サブルーチンレジスタ復帰前とメインルーチン実行後を比較すると、緑枠で囲まれたd0からd7、a1からa6のレジスタが復帰している（サブルーチン実行前と同じ状態になっている）ことが分かる。

(5)

転送後に、転送元の領域を0で上書きする必要があると考える。

なぜなら、「複製」ではなく「転送」であるからである。確かに上記のコードではa1レジスタで指定されるアドレスから始まるメモリの領域とa2レジスタで指定されるアドレスから始まる領域は「複製」された同じ状態になっている。そして、レジスタ退避して転送元の先頭アドレスは実行後には分からない状態にある。しかしながら、メモリを探索すれば転送元のデータは元の状態と変わらないため、「転送」とは言えない。（データを特定のメモリ領域から転送することで削除したつもりでも、メモリアドレスを記憶しているレジスタから削除しただけでメモリから完全に削除しきれていないということ）

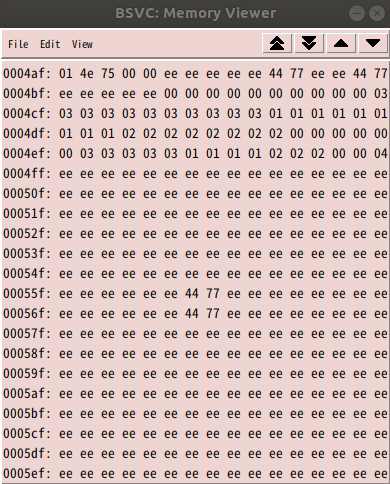
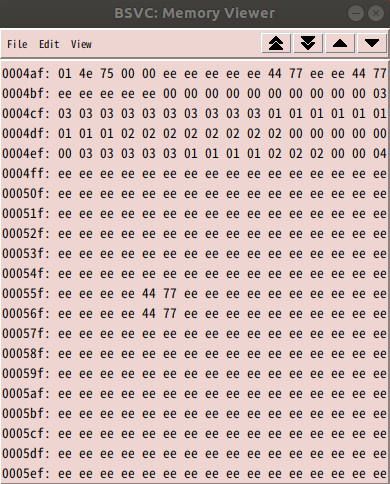
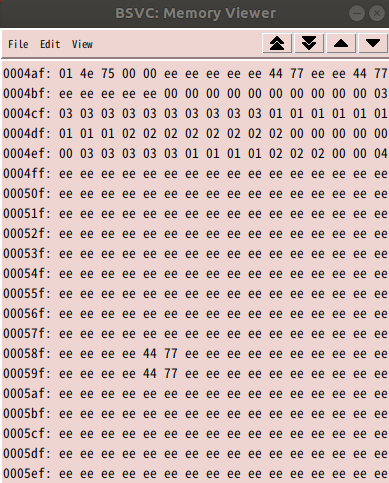
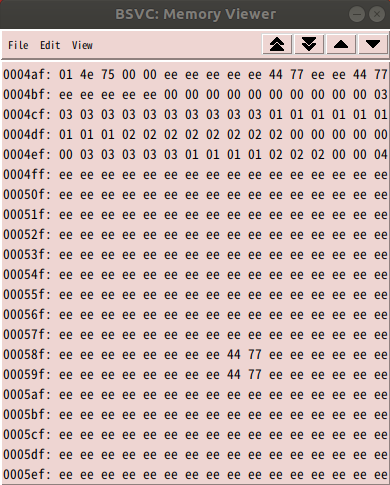
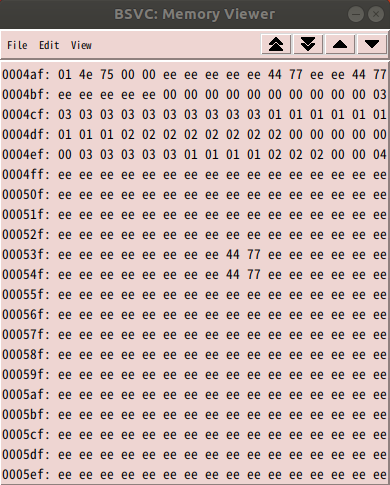
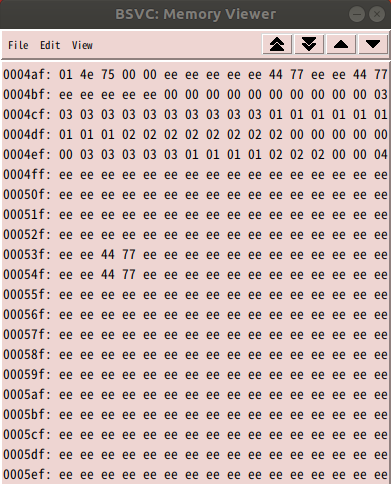
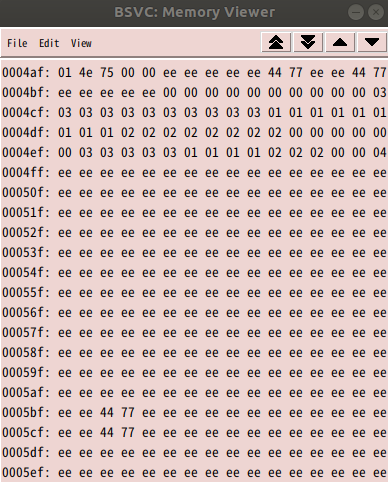
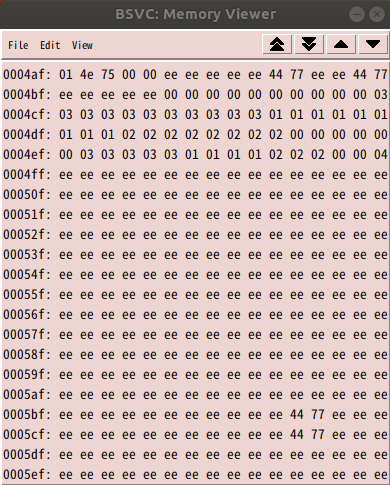
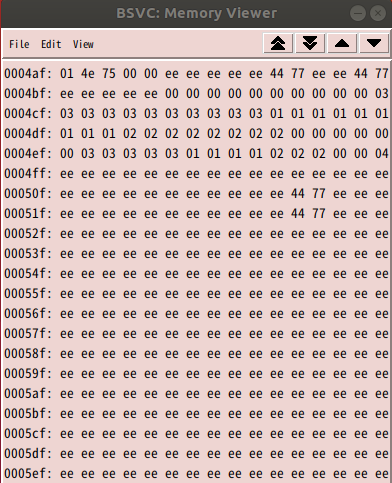
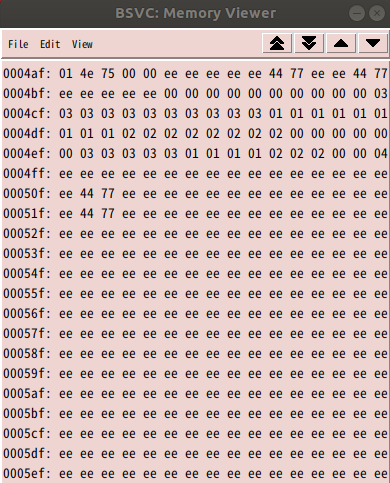
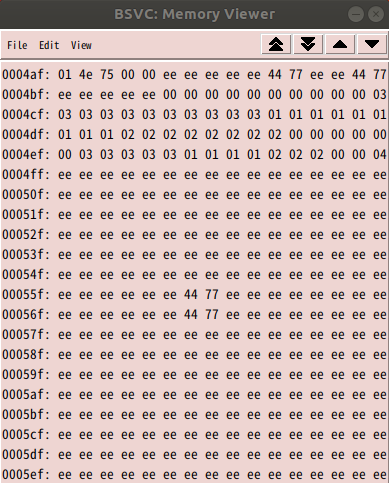
(6)

今後可能なら改良すべき点として、本課題の前提条件であった「アドレスレジスタa1で指定される領域と、アドレスレジスタa2で指定される領域は重ならないものとする」を考慮したプログラムの作成が考えられる。例えば、メモリ全体の大きさの把握し、転送先の領域の確保を転送前に行うなどが挙げられる。

問5

(1)

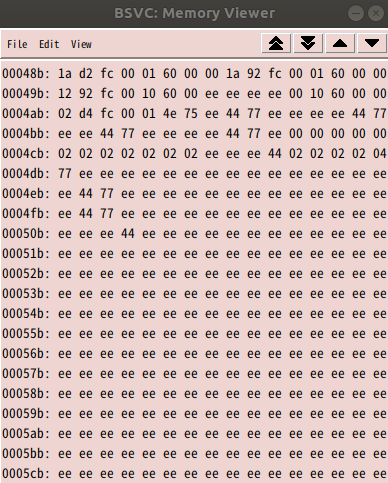
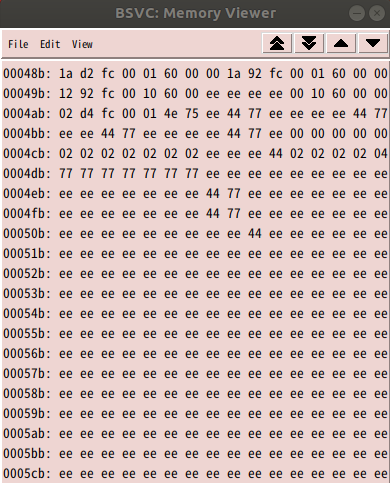
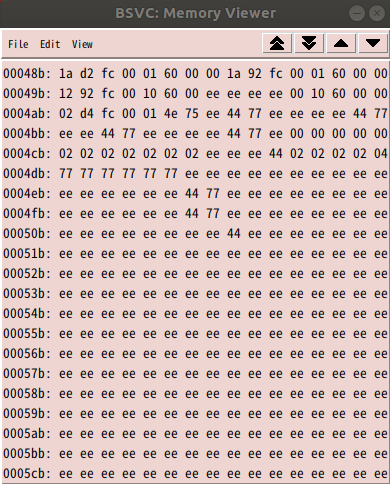
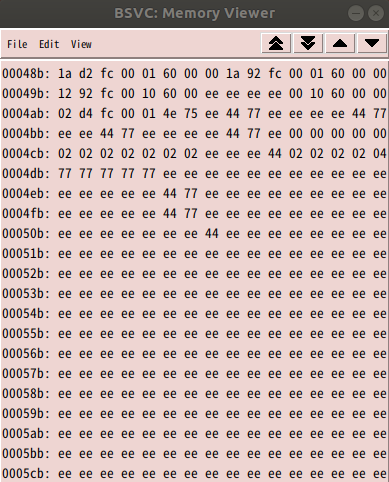
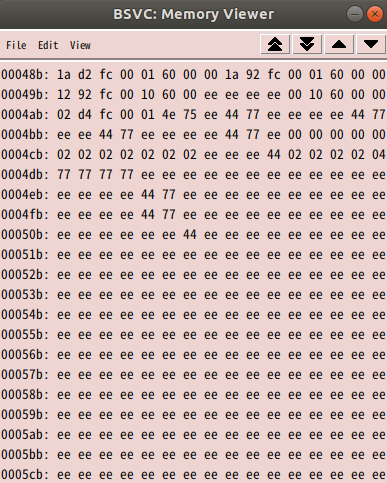
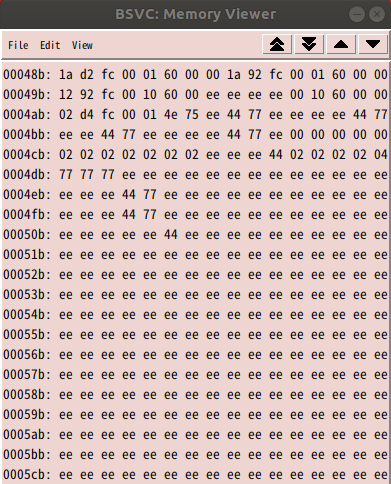
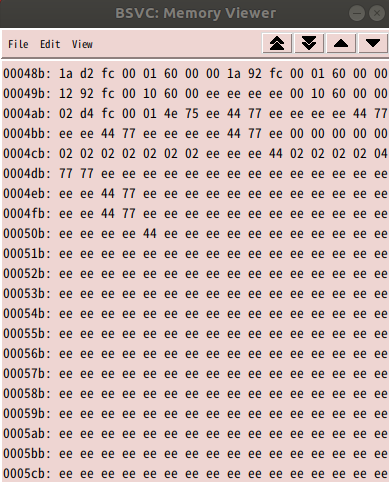
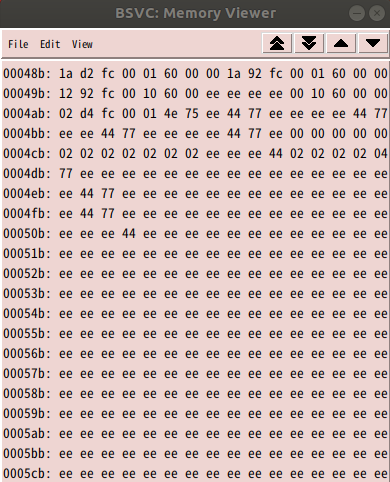
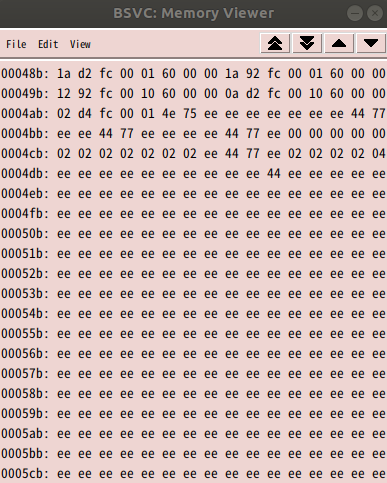
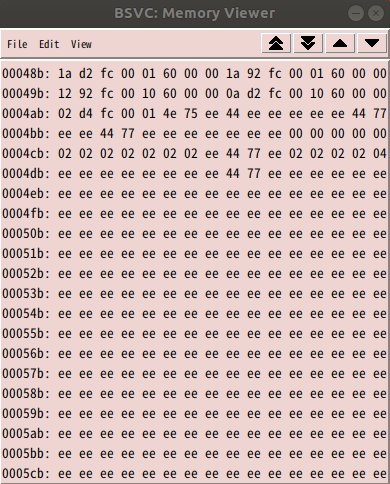
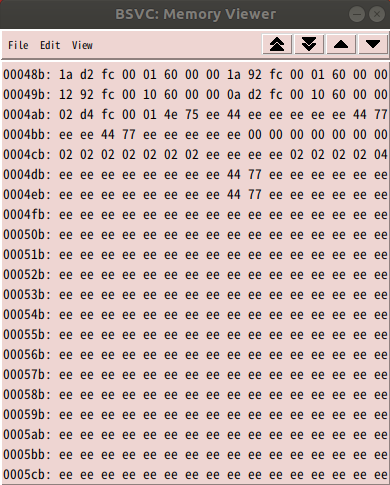
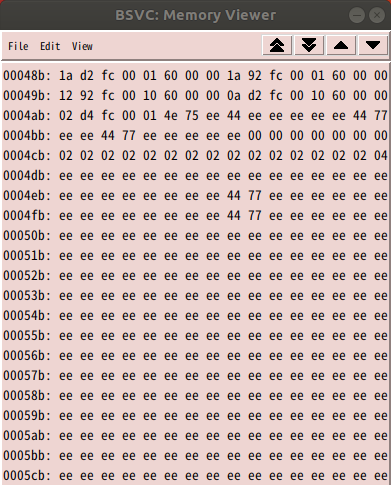
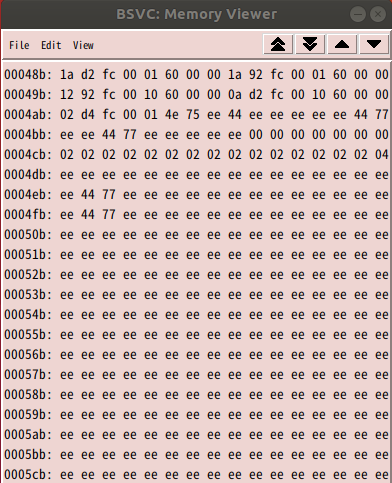
描画する絵を動かす方向を示すデータ領域の「CONTROL」の数値が変わるタイミングを比較することで与えられたプログラムの動作確認を行う。

以上の画面のハードコピー中、緑枠で囲まれた絵が”ee”に埋め尽くされた領域を時計回りに渦を巻くように動いていることが確認できる。

(2)

まず、(1)と同様にメモリの領域の変化を以下に示す。



まず、以上の画面のハードコピー中、青枠で囲まれた絵が(1)同様に描画する絵を動かす方向を示すデータ領域の「CONTROL」の数値に従って動いていることが確認できる。このとき、ハードコピー3枚目から5枚目にかけて、本来描画されるべき範囲である「VIEWTOP」以下の範囲を超えて、他のデータ領域に侵食してしまっている。

3枚目では「CONTROL」の一部を変えているが、次の動きには影響がまだない。

4枚目では実行済みの「CONTROL」の一部を変え、絵となるデータの一部を侵食し始めるが、上書きした部分が”ee”と同じであったため、次の動きに影響はない。

5枚目では絵となるデータの一部を侵食し、異なる値で上書きした。

しかし、6枚目ではここで絵となるデータの先頭アドレスを格納するa1レジスタの値が変わり、絵となるデータの一部を他の値に変え、さらに、データ領域を超えてテキスト領域に侵食してUPとDOWNの一部に影響している。

ゆえに、データ領域において6枚目から12枚目の赤枠に囲まれた部分がRIGHTによって繰り返される。

以上が、「メモリ破壊」の影響と原因である。

(3)

上記のメモリ破壊を防ぐため、描画領域の境界条件として絵となるデータの先頭アドレスを格納するa1レジスタの値と描画領域の大きさを示す#AREASIZEを考慮したプログラムに改良すべきである。