

## Preliminary Report # 2

### Question 1.

The following pseudo code governs the overall aspect of the code written in question 1:

```
CONST NUMVAL = 0x120
CONST FIRST = 0x20000680
DATA_SECTION
  CTR1 ← 0x0B
  MSG ← "Copying table..."
  MSG_ENDLINE ← 0x0D
  MSG_END ← 0x04
```

#### **MAIN\_SECTION**

```
R9 ← 10
R8 ← 0
R4 ← NUMVAL
```

```
WHILE (TRUE)
  R5 = R4 / R9
  R6 = R5 * R9
  R7 = R4 - R6
```

```
IF (R7 == 0) THEN
  PUSH R7
  R8 ← R8 + 1
  R4 ← R5
```

```
ELSE
  PUSH R5
  R8 ← R8 + 1
```

```
IF (R8 == 0) THEN
  BREAK
```

```
R8 ← R8 - 1
R2 ← FIRST
```

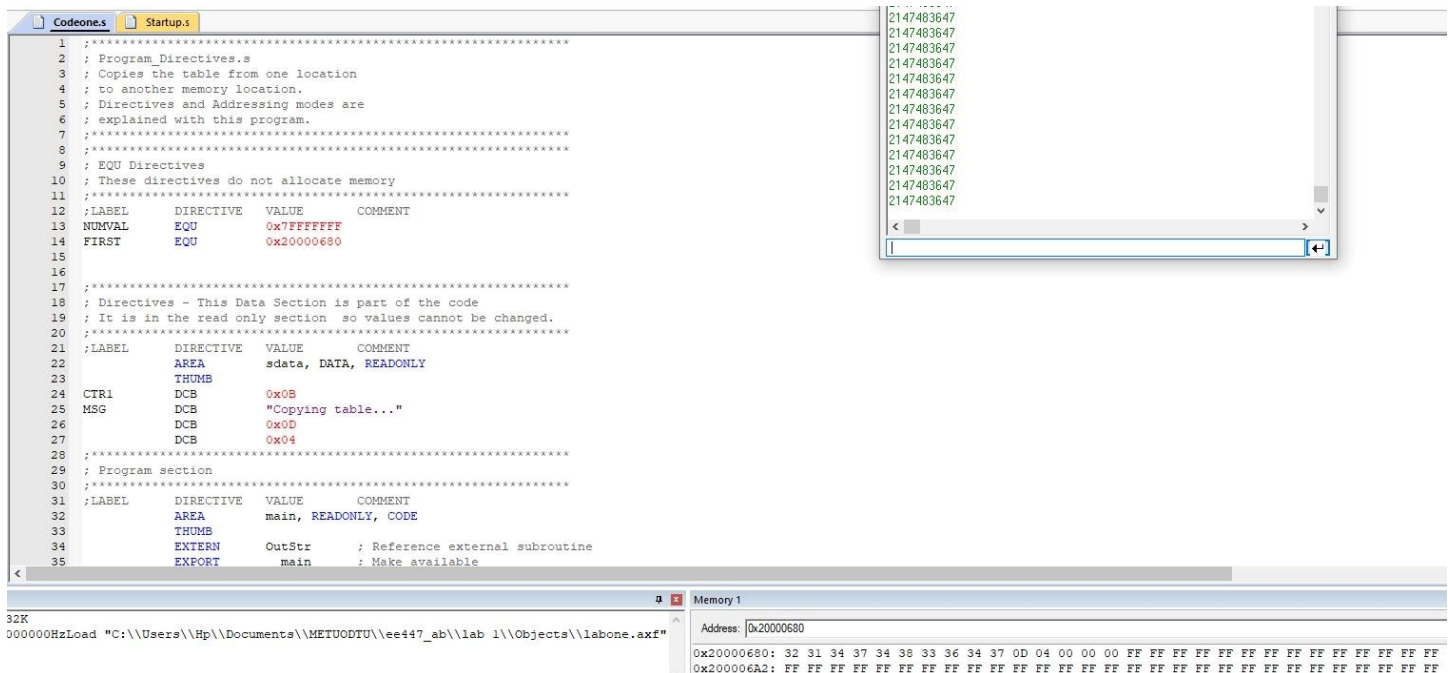
```
WHILE (R8 > 0)
  R1 ← POP()
  R1 ← R1 + 0x30
  [R2] → R1
  R2 ← R2 + 1
  R8 ← R8 - 1
```

```
[R2] → MSG_ENDLINE
R2 ← R2 + 1
R1 ← MSG_END
R1 ← R1 + 0x30
[R2] → R1
R2 ← FIRST
```

```
R0 ← R2
CALL OutStr
CONTINUE
```

## Preliminary Report # 2

The following picture indicates the relevant outputs as shown below, the address indicated by 0x2000.0680 is pointed by R0 and in the termite emulator the max number 0x7FFF.FFFF in decimal can be observed in the display.



The screenshot shows the Codeone.s editor with the following assembly code:

```
1 ; *****
2 ; Program_Directives.s
3 ; Copies the table from one location
4 ; to another memory location.
5 ; Directives and Addressing modes are
6 ; explained with this program.
7 ; *****
8 ; EQU Directives
9 ; These directives do not allocate memory
10 ; *****
11 ; *****
12 ; LABEL DIRECTIVE VALUE COMMENT
13 NUMVAL EQU 0x7FFFFFFF
14 FIRST EQU 0x20000680
15
16 ; *****
17 ; Directives - This Data Section is part of the code
18 ; It is in the read only section so values cannot be changed.
19 ; *****
20 ; *****
21 ; LABEL DIRECTIVE VALUE COMMENT
22 AREA sdata, DATA, READONLY
23 THUMB
24 CTRL DCB 0x0B
25 MSG DCB "Copying table..."
26 DCB 0x0D
27 DCB 0x04
28 ; *****
29 ; Program section
30 ; *****
31 ; LABEL DIRECTIVE VALUE COMMENT
32 AREA main, READONLY, CODE
33 THUMB
34 EXTERN OutStr ; Reference external subroutine
35 EXPORT main ; Make available
```

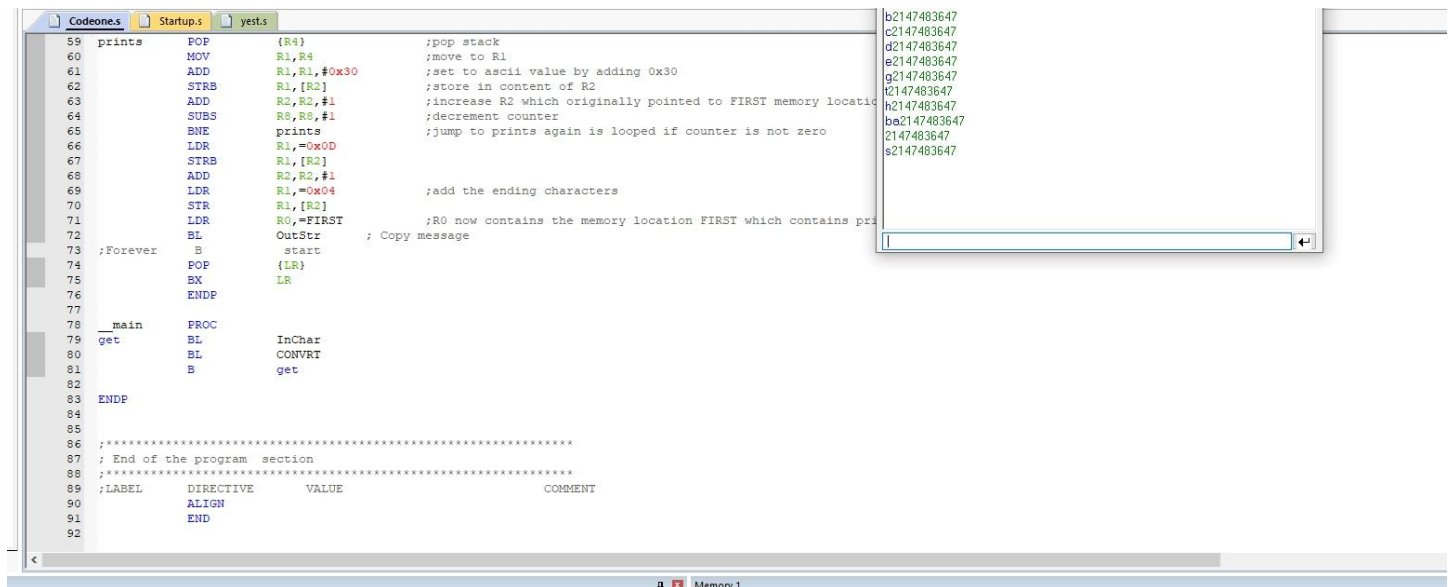
The memory dump window shows the following data:

Address	0x20000680
0x20000680: 32 31 34 37 34 38 33 36 34 37 0D 04 00 00 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	
0x200006A2: FF	

Fig 1. Output Results for Q1

Question 2.

In this program both the sub routines of CONVRT and InChar have been implemented as shown and the emulator output has been observed as follows as shown in the diagram.



The screenshot shows the Codeone.s editor with the following assembly code:

```
59 prints POP {R4} ;POP stack
60 MOV R1,R4 ;move to R1
61 ADD R1,R1,#0x30 ;set to ascii value by adding 0x30
62 STRB R1,[R2] ;store in content of R2
63 ADD R2,R2,#1 ;increase R2 which originally pointed to FIRST memory location
64 SUBS R8,R8,#1 ;decrement counter
65 BNE prints ;jump to prints again is looped if counter is not zero
66 LDR R1,#0x0D
67 STRB R1,[R2]
68 ADD R2,R2,#1
69 LDR R1,#0x04
70 STR R1,[R2]
71 LDR R0,FIRST ;R0 now contains the memory location FIRST which contains print message
72 BL OutStr ; Copy message
73 ;Forever B start
74 POP {LR}
75 BX LR
76 ENDP
77
78 _main PROC
79 get BL InChar
80 BL CONVRT
81 B get
82
83 ENDP
84
85 ; *****
86 ; End of the program section
87 ; *****
88 ; *****
89 ; LABEL DIRECTIVE VALUE COMMENT
90 ALIGN
91 END
```

The memory dump window shows the following data:

Address	0x20000680
0x20000680: b2147483647 c2147483647 d2147483647 e2147483647 g2147483647 h2147483647 i2147483647 j2147483647 k2147483647 l2147483647 m2147483647 n2147483647 o2147483647 p2147483647 q2147483647 r2147483647 s2147483647 t2147483647 u2147483647 v2147483647 w2147483647 x2147483647 y2147483647 z2147483647	

Fig 2. Output Results for Q2

## Preliminary Report # 2

### Question 3.

The following flow chart shows the path flow of data while using the subroutine functions of InChar OutChar, CONVRT and UPBND incorporated in the diagram:

The flowchart illustrates a number-guessing program that starts by prompting the user to input 'n', representing the maximum range for the number to be guessed (MAX\_N). It initializes the lower (R1) and upper (R2) boundaries and the current guess (R3). The program enters a loop where it displays the current guess, awaits user feedback ('C' for correct, 'U' for up, 'D' for down), and refines the boundaries accordingly. If the user input is 'C', indicating a correct guess, the program ends. If 'U' is received, the upper boundary is adjusted, and if 'D' is received, the lower boundary is adapted. This loop continues until the correct number is identified.

```
Disassembly
Reset_Handler:
00000000 200:  B       0x00000098 __main
00000001 201:  BUMP
00000002 202:  BUMP
00000003 203:  BUMP
00000004 204:  BUMP
00000005 205:  BUMP
00000006 206:  BUMP
00000007 207:  BUMP
00000008 208:  BUMP
00000009 209:  BUMP
0000000A 210:  BUMP
0000000B 211:  BUMP
0000000C 212:  BUMP
0000000D 213:  BUMP
0000000E 214:  BUMP
0000000F 215:  BUMP
00000010 216:  BUMP
00000011 217:  BUMP
00000012 218:  BUMP
00000013 219:  BUMP
00000014 220:  BUMP
00000015 221:  BUMP
00000016 222:  BUMP
00000017 223:  BUMP
00000018 224:  BUMP
00000019 225:  BUMP
0000001A 226:  BUMP
0000001B 227:  BUMP
0000001C 228:  BUMP
0000001D 229:  BUMP
0000001E 230:  BUMP
0000001F 231:  BUMP
00000020 232:  BUMP
00000021 233:  BUMP
00000022 234:  BUMP
00000023 235:  BUMP
00000024 236:  BUMP
00000025 237:  BUMP
00000026 238:  BUMP
00000027 239:  BUMP
00000028 240:  BUMP
00000029 241:  BUMP
0000002A 242:  BUMP
0000002B 243:  BUMP
0000002C 244:  BUMP
0000002D 245:  BUMP
0000002E 246:  BUMP
0000002F 247:  BUMP
00000030 248:  BUMP
00000031 249:  BUMP
00000032 250:  BUMP
00000033 251:  BUMP
00000034 252:  BUMP
00000035 253:  BUMP
00000036 254:  BUMP
00000037 255:  BUMP
00000038 256:  BUMP
00000039 257:  BUMP
0000003A 258:  BUMP
0000003B 259:  BUMP
0000003C 260:  BUMP
0000003D 261:  BUMP
0000003E 262:  BUMP
0000003F 263:  BUMP
00000040 264:  BUMP
00000041 265:  BUMP
00000042 266:  BUMP
00000043 267:  BUMP
00000044 268:  BUMP
00000045 269:  BUMP
00000046 270:  BUMP
00000047 271:  BUMP
00000048 272:  BUMP
00000049 273:  BUMP
0000004A 274:  BUMP
0000004B 275:  BUMP
0000004C 276:  BUMP
0000004D 277:  BUMP
0000004E 278:  BUMP
0000004F 279:  BUMP
00000050 280:  BUMP
00000051 281:  BUMP
00000052 282:  BUMP
00000053 283:  BUMP
00000054 284:  BUMP
00000055 285:  BUMP
00000056 286:  BUMP
00000057 287:  BUMP
00000058 288:  BUMP
00000059 289:  BUMP
0000005A 290:  BUMP
0000005B 291:  BUMP
0000005C 292:  BUMP
0000005D 293:  BUMP
0000005E 294:  BUMP
0000005F 295:  BUMP
00000060 296:  BUMP
00000061 297:  BUMP
00000062 298:  BUMP
00000063 299:  BUMP
00000064 300:  BUMP
00000065 301:  BUMP
00000066 302:  BUMP
00000067 303:  BUMP
00000068 304:  BUMP
00000069 305:  BUMP
0000006A 306:  BUMP
0000006B 307:  BUMP
0000006C 308:  BUMP
0000006D 309:  BUMP
0000006E 310:  BUMP
0000006F 311:  BUMP
00000070 312:  BUMP
00000071 313:  BUMP
00000072 314:  BUMP
00000073 315:  BUMP
00000074 316:  BUMP
00000075 317:  BUMP
00000076 318:  BUMP
00000077 319:  BUMP
00000078 320:  BUMP
00000079 321:  BUMP
0000007A 322:  BUMP
0000007B 323:  BUMP
0000007C 324:  BUMP
0000007D 325:  BUMP
0000007E 326:  BUMP
0000007F 327:  BUMP
00000080 328:  BUMP
00000081 329:  BUMP
00000082 330:  BUMP
00000083 331:  BUMP
00000084 332:  BUMP
00000085 333:  BUMP
00000086 334:  BUMP
00000087 335:  BUMP
00000088 336:  BUMP
00000089 337:  BUMP
0000008A 338:  BUMP
0000008B 339:  BUMP
0000008C 340:  BUMP
0000008D 341:  BUMP
0000008E 342:  BUMP
0000008F 343:  BUMP
00000090 344:  BUMP
00000091 345:  BUMP
00000092 346:  BUMP
00000093 347:  BUMP
00000094 348:  BUMP
00000095 349:  BUMP
00000096 350:  BUMP
00000097 351:  BUMP
00000098 352:  BUMP
00000099 353:  BUMP
0000009A 354:  BUMP
0000009B 355:  BUMP
0000009C 356:  BUMP
0000009D 357:  BUMP
0000009E 358:  BUMP
0000009F 359:  BUMP
000000A0 360:  BUMP
000000A1 361:  BUMP
000000A2 362:  BUMP
000000A3 363:  BUMP
000000A4 364:  BUMP
000000A5 365:  BUMP
000000A6 366:  BUMP
000000A7 367:  BUMP
000000A8 368:  BUMP
000000A9 369:  BUMP
000000AA 370:  BUMP
000000AB 371:  BUMP
000000AC 372:  BUMP
000000AD 373:  BUMP
000000AE 374:  BUMP
000000AF 375:  BUMP
000000B0 376:  BUMP
000000B1 377:  BUMP
000000B2 378:  BUMP
000000B3 379:  BUMP
000000B4 380:  BUMP
000000B5 381:  BUMP
000000B6 382:  BUMP
000000B7 383:  BUMP
000000B8 384:  BUMP
000000B9 385:  BUMP
000000BA 386:  BUMP
000000BB 387:  BUMP
000000BC 388:  BUMP
000000BD 389:  BUMP
000000BE 390:  BUMP
000000BF 391:  BUMP
000000C0 392:  BUMP
000000C1 393:  BUMP
000000C2 394:  BUMP
000000C3 395:  BUMP
000000C4 396:  BUMP
000000C5 397:  BUMP
000000C6 398:  BUMP
000000C7 399:  BUMP
000000C8 400:  BUMP
000000C9 401:  BUMP
000000CA 402:  BUMP
000000CB 403:  BUMP
000000CC 404:  BUMP
000000CD 405:  BUMP
000000CE 406:  BUMP
000000CF 407:  BUMP
000000D0 408:  BUMP
000000D1 409:  BUMP
000000D2 410:  BUMP
000000D3 411:  BUMP
000000D4 412:  BUMP
000000D5 413:  BUMP
000000D6 414:  BUMP
000000D7 415:  BUMP
000000D8 416:  BUMP
000000D9 417:  BUMP
000000DA 418:  BUMP
000000DB 419:  BUMP
000000DC 420:  BUMP
000000DD 421:  BUMP
000000DE 422:  BUMP
000000DF 423:  BUMP
000000E0 424:  BUMP
000000E1 425:  BUMP
000000E2 426:  BUMP
000000E3 427:  BUMP
000000E4 428:  BUMP
000000E5 429:  BUMP
000000E6 430:  BUMP
000000E7 431:  BUMP
000000E8 432:  BUMP
000000E9 433:  BUMP
000000EA 434:  BUMP
000000EB 435:  BUMP
000000EC 436:  BUMP
000000ED 437:  BUMP
000000EE 438:  BUMP
000000EF 439:  BUMP
000000F0 440:  BUMP
000000F1 441:  BUMP
000000F2 442:  BUMP
000000F3 443:  BUMP
000000F4 444:  BUMP
000000F5 445:  BUMP
000000F6 446:  BUMP
000000F7 447:  BUMP
000000F8 448:  BUMP
000000F9 449:  BUMP
000000FA 450:  BUMP
000000FB 451:  BUMP
000000FC 452:  BUMP
000000FD 453:  BUMP
000000FE 454:  BUMP
000000FF 455:  BUMP
00000100 456:  BUMP
00000101 457:  BUMP
00000102 458:  BUMP
00000103 459:  BUMP
00000104 460:  BUMP
00000105 461:  BUMP
00000106 462:  BUMP
00000107 463:  BUMP
00000108 464:  BUMP
00000109 465:  BUMP
0000010A 466:  BUMP
0000010B 467:  BUMP
0000010C 468:  BUMP
0000010D 469:  BUMP
0000010E 470:  BUMP
0000010F 471:  BUMP
00000110 472:  BUMP
00000111 473:  BUMP
00000112 474:  BUMP
00000113 475:  BUMP
00000114 476:  BUMP
00000115 477:  BUMP
00000116 478:  BUMP
00000117 479:  BUMP
00000118 480:  BUMP
00000119 481:  BUMP
0000011A 482:  BUMP
0000011B 483:  BUMP
0000011C 484:  BUMP
0000011D 485:  BUMP
0000011E 486:  BUMP
0000011F 487:  BUMP
00000120 488:  BUMP
00000121 489:  BUMP
00000122 490:  BUMP
00000123 491:  BUMP
00000124 492:  BUMP
00000125 493:  BUMP
00000126 494:  BUMP
00000127 495:  BUMP
00000128 496:  BUMP
00000129 497:  BUMP
0000012A 498:  BUMP
0000012B 499:  BUMP
0000012C 500:  BUMP
0000012D 501:  BUMP
0000012E 502:  BUMP
0000012F 503:  BUMP
00000130 504:  BUMP
00000131 505:  BUMP
00000132 506:  BUMP
00000133 507:  BUMP
00000134 508:  BUMP
00000135 509:  BUMP
00000136 510:  BUMP
00000137 511:  BUMP
00000138 512:  BUMP
00000139 513:  BUMP
0000013A 514:  BUMP
0000013B 515:  BUMP
0000013C 516:  BUMP
0000013D 517:  BUMP
0000013E 518:  BUMP
0000013F 519:  BUMP
00000140 520:  BUMP
00000141 521:  BUMP
00000142 522:  BUMP
00000143 523:  BUMP
00000144 524:  BUMP
00000145 525:  BUMP
00000146 526:  BUMP
00000147 527:  BUMP
00000148 528:  BUMP
00000149 529:  BUMP
0000014A 530:  BUMP
0000014B 531:  BUMP
0000014C 532:  BUMP
0000014D 533:  BUMP
0000014E 534:  BUMP
0000014F 535:  BUMP
00000150 536:  BUMP
00000151 537:  BUMP
00000152 538:  BUMP
00000153 539:  BUMP
00000154 540:  BUMP
00000155 541:  BUMP
00000156 542:  BUMP
00000157 543:  BUMP
00000158 544:  BUMP
00000159 545:  BUMP
0000015A 546:  BUMP
0000015B 547:  BUMP
0000015C 548:  BUMP
0000015D 549:  BUMP
0000015E 550:  BUMP
0000015F 551:  BUMP
00000160 552:  BUMP
00000161 553:  BUMP
00000162 554:  BUMP
00000163 555:  BUMP
00000164 556:  BUMP
00000165 557:  BUMP
00000166 558:  BUMP
00000167 559:  BUMP
00000168 560:  BUMP
00000169 561:  BUMP
0000016A 562:  BUMP
0000016B 563:  BUMP
0000016C 564:  BUMP
0000016D 565:  BUMP
0000016E 566:  BUMP
0000016F 567:  BUMP
00000170 568:  BUMP
00000171 569:  BUMP
00000172 570:  BUMP
00000173 571:  BUMP
00000174 572:  BUMP
00000175 573:  BUMP
00000176 574:  BUMP
00000177 575:  BUMP
00000178 576:  BUMP
00000179 577:  BUMP
0000017A 578:  BUMP
0000017B 579:  BUMP
0000017C 580:  BUMP
0000017D 581:  BUMP
0000017E 582:  BUMP
0000017F 583:  BUMP
00000180 584:  BUMP
00000181 585:  BUMP
00000182 586:  BUMP
00000183 587:  BUMP
00000184 588:  BUMP
00000185 589:  BUMP
00000186 590:  BUMP
00000187 591:  BUMP
00000188 592:  BUMP
00000189 593:  BUMP
0000018A 594:  BUMP
0000018B 595:  BUMP
0000018C 596:  BUMP
0000018D 597:  BUMP
0000018E 598:  BUMP
0000018F 599:  BUMP
00000190 600:  BUMP
00000191 601:  BUMP
00000192 602:  BUMP
00000193 603:  BUMP
00000194 604:  BUMP
00000195 605:  BUMP
00000196 606:  BUMP
00000197 607:  BUMP
00000198 608:  BUMP
00000199 609:  BUMP
0000019A 610:  BUMP
0000019B 611:  BUMP
0000019C 612:  BUMP
0000019D 613:  BUMP
0000019E 614:  BUMP
0000019F 615:  BUMP
000001A0 616:  BUMP
000001A1 617:  BUMP
000001A2 618:  BUMP
000001A3 619:  BUMP
000001A4 620:  BUMP
000001A5 621:  BUMP
000001A6 622:  BUMP
000001A7 623:  BUMP
000001A8 624:  BUMP
000001A9 625:  BUMP
000001AA 626:  BUMP
000001AB 627:  BUMP
000001AC 628:  BUMP
000001AD 629:  BUMP
000001AE 630:  BUMP
000001AF 631:  BUMP
000001B0 632:  BUMP
000001B1 633:  BUMP
000001B2 634:  BUMP
000001B3 635:  BUMP
000001B4 636:  BUMP
000001B5 637:  BUMP
000001B6 638:  BUMP
000001B7 639:  BUMP
000001B8 640:  BUMP
000001B9 641:  BUMP
000001BA 642:  BUMP
000001BB 643:  BUMP
000001BC 644:  BUMP
000001BD 645:  BUMP
000001BE 646:  BUMP
000001BF 647:  BUMP
000001C0 648:  BUMP
000001C1 649:  BUMP
000001C2 650:  BUMP
000001C3 651:  BUMP
000001C4 652:  BUMP
000001C5 653:  BUMP
000001C6 654:  BUMP
000001C7 655:  BUMP
000001C8 656:  BUMP
000001C9 657:  BUMP
000001CA 658:  BUMP
000001CB 659:  BUMP
000001CC 660:  BUMP
000001CD 661:  BUMP
000001CE 662:  BUMP
000001CF 663:  BUMP
000001D0 664:  BUMP
000001D1 665:  BUMP
000001D2 666:  BUMP
000001D3 667:  BUMP
000001D4 668:  BUMP
000001D5 669:  BUMP
000001D6 670:  BUMP
000001D7 671:  BUMP
000001D8 672:  BUMP
000001D9 673:  BUMP
000001DA 674:  BUMP
000001DB 675:  BUMP
000001DC 676:  BUMP
000001DD 677:  BUMP
000001DE 678:  BUMP
000001DF 679:  BUMP
000001E0 680:  BUMP
000001E1 681:  BUMP
000001E2 682:  BUMP
000001E3 683:  BUMP
000001E4 684:  BUMP
000001E5 685:  BUMP
000001E6 686:  BUMP
000001E7 687:  BUMP
000001E8 688:  BUMP
000001E9 689:  BUMP
000001EA 690:  BUMP
000001EB 691:  BUMP
000001EC 692:  BUMP
000001ED 693:  BUMP
000001EE 694:  BUMP
000001EF 695:  BUMP
000001F0 696:  BUMP
000001F1 697:  BUMP
000001F2 698:  BUMP
000001F3 699:  BUMP
000001F4 700:  BUMP
000001F5 701:  BUMP
000001F6 702:  BUMP
000001F7 703:  BUMP
000001F8 704:  BUMP
000001F9 705:  BUMP
000001FA 706:  BUMP
000001FB 707:  BUMP
000001FC 708:  BUMP
000001FD 709:  BUMP
000001FE 710:  BUMP
000001FF 711:  BUMP
00000200 712:  BUMP
00000201 713:  BUMP
00000202 714:  BUMP
00000203 715:  BUMP
00000204 716:  BUMP
00000205 717:  BUMP
00000206 718:  BUMP
00000207 719:  BUMP
00000208 720:  BUMP
00000209 721:  BUMP
0000020A 722:  BUMP
0000020B 723:  BUMP
0000020C 724:  BUMP
0000020D 725:  BUMP
0000020E 726:  BUMP
0000020F 727:  BUMP
00000210 728:  BUMP
00000211 729:  BUMP
00000212 730:  BUMP
00000213 731:  BUMP
00000214 732:  BUMP
00000215 733:  BUMP
00000216 734:  BUMP
00000217 735:  BUMP
00000218 736:  BUMP
00000219 737:  BUMP
0000021A 738:  BUMP
0000021B 739:  BUMP
0000021C 740:  BUMP
0000021D 741:  BUMP
0000021E 742:  BUMP
0000021F 743:  BUMP
00000220 744:  BUMP
00000221 745:  BUMP
00000222 746:  BUMP
00000223 747:  BUMP
00000224 748:  BUMP
00000225 749:  BUMP
00000226 750:  BUMP
00000227 751:  BUMP
00000228 752:  BUMP
00000229 753:  BUMP
0000022A 754:  BUMP
0000022B 755:  BUMP
0000022C 756:  BUMP
0000022D 757:  BUMP
0000022E 758:  BUMP
0000022F 759:  BUMP
00000230 760:  BUMP
00000231 761:  BUMP
00000232 762:  BUMP
00000233 763:  BUMP
00000234 764:  BUMP
00000235 765:  BUMP
00000236 766:  BUMP
00000237 767:  BUMP
00000238 768:  BUMP
00000239 769:  BUMP
0000023A 770:  BUMP
0000023B 771:  BUMP
0000023C 772:  BUMP
0000023D 773:  BUMP
0000023E 774:  BUMP
0000023F 775:  BUMP
00000240 776:  BUMP
00000241 777:  BUMP
00000242 778:  BUMP
00000243 779:  BUMP
00000244 780:  BUMP
00000245 781:  BUMP
00000246 782:  BUMP
00000247 783:  BUMP
00000248 784:  BUMP
00000249 785:  BUMP
0000024A 786:  BUMP
0000024B 787:  BUMP
0000024C 788:  BUMP
0000024D 789:  BUMP
0000024E 790:  BUMP
0000024F 791:  BUMP
00000250 792:  BUMP
00000251 793:  BUMP
00000252 794:  BUMP
00000253 795:  BUMP
00000254 796:  BUMP
00000255 797:  BUMP
00000256 798:  BUMP
00000257 799:  BUMP
00000258 800:  BUMP
00000259 801:  BUMP
0000025A 802:  BUMP
0000025B 803:  BUMP
0000025C 804:  BUMP
0000025D 805:  BUMP
0000025E 806:  BUMP
0000025F 807:  BUMP
00000260 808:  BUMP
00000261 809:  BUMP
00000262 810:  BUMP
00000263 811:  BUMP
00000264 812:  BUMP
00000265 813:  BUMP
00000266 814:  BUMP
00000267 815:  BUMP
00000268 816:  BUMP
00000269 817:  BUMP
0000026A 818:  BUMP
0000026B 819:  BUMP
0000026C 820:  BUMP
0000026D 821:  BUMP
0000026E 822:  BUMP
0000026F 823:  BUMP
00000270 824:  BUMP
00000271 825:  BUMP
00000272 826:  BUMP
00000273 827:  BUMP
00000274 828:  BUMP
00000275 829:  BUMP
00000276 830:  BUMP
00000277 831:  BUMP
00000278 832:  BUMP
00000279 833:  BUMP
0000027A 834:  BUMP
0000027B 835:  BUMP
0000027C 836:  BUMP
0000027D 837:  BUMP
0000027E 838:  BUMP
0000027F 839:  BUMP
00000280 840:  BUMP
00000281 841:  BUMP
00000282 842:  BUMP
00000283 843:  BUMP
00000284 844:  BUMP
00000285 845:  BUMP
00000286 846:  BUMP
00000287 847:  BUMP
00000288 848:  BUMP
00000289 849:  BUMP
0000028A 850:  BUMP
0000028B 851:  BUMP
0000028C 852:  BUMP
0000028D 853:  BUMP
0000028E 854:  BUMP
0000028F 855:  BUMP
00000290 856:  BUMP
00000291 857:  BUMP
00000292 858:  BUMP
00000293 859:  BUMP
00000294 860:  BUMP
00000295 861:  BUMP
00000296 862:  BUMP
00000297 863:  BUMP
00000298 864:  BUMP
00000299 865:  BUMP
0000029A 866:  BUMP
0000029B 867:  BUMP
0000029C 868:  BUMP
0000029D 869:  BUMP
0000029E 870:  BUMP
0000029F 871:  BUMP
000002A0 872:  BUMP
000002A1 873:  BUMP
000002A2 874:  BUMP
000002A3 875:  BUMP
000002A4 876:  BUMP
000002A5 877:  BUMP
000002A6 878:  BUMP
000002A7 879:  BUMP
000002A8 880:  BUMP
000002A9 881:  BUMP
000002AA 882:  BUMP
000002AB 883:  BUMP
000002AC 884:  BUMP
000002AD 885:  BUMP
000002AE 886:  BUMP
000002AF 887:  BUMP
000002B0 888:  BUMP
000002B1 889:  BUMP
000002B2 890:  BUMP
000002B3 891:  BUMP
000002B4 892:  BUMP
000002B5 893:  BUMP
000002B6 894:  BUMP
000002B7 895:  BUMP
000002B8 896:  BUMP
000002B9 897:  BUMP
000002BA 898:  BUMP
000002BB 899:  BUMP
000002BC 900:  BUMP
000002BD 901:  BUMP
000002BE 902:  BUMP
000002BF 903:  BUMP
000002C0 904:  BUMP
000002C
```

## Preliminary Report # 2

### Question 4.

This pseudo code program gives us the control algorithm for the monster game:

Start

*Input: Monster's health, Sword damage*

*if Monster's health  $\leq 0$  then*

*Monster is defeated*

*Print Monster's health (after-heal health)*

*Return*

*else*

*Calculate new Monster's health after the strike*

*if Monster's health is now negative then*

*Calculate healing amount as half of the initial health*

*Add the healing amount to Monster's health*

*Halve Sword damage*

*end if*

*Push initial health onto the stack*

*Recursively call the function with updated health and damage*

*Pop initial health from the stack*

*Print Monster's health (after-heal health)*

*Print Initial health*

*end if*

End

The following diagram shows the respective outputs and their input conditions.

The screenshot displays a code editor with assembly code for a monster game. The code is organized into sections: **BATTLE**, **MNS\_REC**, and **HEALTH\_OUT**. The **BATTLE** section checks if the monster's health is less than or equal to zero. If so, it goes to the **HEALTH\_OUT** section. Otherwise, it calculates the new health after a strike, checks if it's negative, and if so, calculates a healing amount based on the initial health. The **MNS\_REC** section pushes the new health onto the stack and increments a counter. The **HEALTH\_OUT** section pops the initial health from the stack and prints it. The terminal window on the right shows the program's output, which includes the initial health (265527), the sword damage (007), and the monster's health after the strike (265527).

```
292: ;*****
133      PUSH    (R3)          ; Load SP with Initial Monster Health
134      LSR     R3, #1         ; Divide Monster health by 2
135      MOV     R8, #1         ; Stack counter
136
137  BATTLE  CMP     R5, #0      ; Check Monster Health to see if it is defeated
138          MOV     R12, R8
139          BLT     MNS_REC     ; If health < 0 then goto MNS_REC
140          BEQ     HEALTH_OUT  ; If health = 0 then goto health out
141          SUB     R5, R6      ; Reduce health by data in R2
142          B       BATTLE     ; loop again until breaks
143
144  MNS_REC  ADD     R5, R5, R3  ; Add half health stoerd in R3 to R1
145          PUSH    (R5)       ; Push new health to SP
146          ADD     R8, #1      ; Increment Counter to update stack
147          LSR     R6, #1      ; reduce sword damage / 2
148          B       BATTLE     ; do battle again
149
150
151  HEALTH_OUT  POP     (R1)
152              BL     CONVRT
153              SUBS    R12, R12, #1
154              BNE     HEALTH_OUT
155              BEQ     Done
156
157  Done     B       Done
158
159
160
161 ;*****
162 ; End of the program section
163 ;*****
164 ; LABEL    DIRECTIVE    VALUE    COMMENT
165          ALIGN
166          END
```

007 enter sword damage  
022  
7  
265527  
IEpW4'INEJHtNq007 enter sword damage  
022  
7  
265527  
enter monster health  
007 enter sword damage  
022  
7  
enter monster health  
007 enter sword damage  
022  
7  
|