

1. 2016-2

```
001 /* (c) 2016 Rahmat M. Samik-Ibrahim -- This is free software
005  * Assume (&ptrchr is 0x7FFFEEDDCCBB, order of bytes: little-endian) */
009 #define LINES 3
010 #include <stdio.h>
012 void printeq(int lines) {
013     while (lines-- > 0 ) printf("= = ");
014     printf("\n");
015 }
017 void main(void) {
018     int ii;
019     unsigned char dummy = 'a';
020     unsigned char* ptrchr = &dummy;
022     printeq(LINES);
023     printf(" dummy: %c\n", dummy);
024     printf("*ptrchr: %c\n", *ptrchr);
025     printeq(LINES);
026     printf("%p\n", &ptrchr);
027     printeq(LINES);
028     ptrchr = (char*) &ptrchr;
029     for (ii=0; ii<6; ii++) {
030         printf("%X ", *ptrchr);
031         ptrchr++;
032     }
033     putchar('\n');
034     printeq(LINES);
035 }
```

(a) Write down the output of this program

2. 2017-1

C Programing	
001 /* 002 * (c) 2017 Rahmat M. Samik-Ibrahim 003 * -- This is free software 004 * REV00 Thu Mar 30 18:27:30 WIB 2017 005 * START Thu Mar 30 18:27:30 WIB 2017 006 * INT is 32 bit little endian 007 * 41H='A'; 42H='B'; 43H='C'; 44H='D' 008 */ 009 #include <stdio.h> 010 char chrarry[]="ZZZZ ZZZZ ";	011 void main(void) { 012 char chrvar = 'M'; 013 int intvar = 0x41424344; 014 int* intptr = (int*) chrarry; 015 printf("YY. chrarry=%p\n", chrarry); 016 printf("ZZ. intprt=%p\n", intptr); 017 printf("01. chrvar=%c\n", chrvar); 018 printf("02. *chrarry=%c\n", *chrarry); 019 printf("03. str chrarry=%s\n", chrarry); 020 *intptr = intvar; 021 printf("04. str chrarry=%s\n", chrarry); 022 }
Program Output (Line: 015, 016, 017, 018, 019, 021):	

3. 2017-2

C Programing ADDR	
001 /* 002 * (c) 2017 Rahmat M. Samik-Ibrahim 003 * http://rahmatm.samik-ibrahim.vlsm.org/ 004 * This is free software. 005 * REV00 Mon Oct 16 21:15:03 WIB 2017 006 * START Mon Oct 16 21:15:03 WIB 2017 007 */ 008 009 #include <stdio.h> 010 011 char* stringChar="HALLO"; 012 char* stringPTR;	014 void main (void) { 015 stringPTR=stringChar; 016 printf ("ADDR1: %p VAL: %p STR: %s\n", &stringChar, 017 stringChar, stringChar); 018 printf ("ADDR2: %p VAL: %p STR: %s\n", &stringPTR, 019 stringPTR, stringPTR); 020 while (*(++stringPTR) != 0) { 021 printf ("ADDR3: %p VAL: %p STR: %s\n", &stringPTR, 022 stringPTR, stringPTR); 023 } 024 }
Program Output:	
ADDR1: 0x601038 VAL: 0x400674 STR: HALLO	
ADDR2: 0x601048 VAL: 0x400674 STR: HALLO	