







|  |
| --- |
| */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  *\* Author: 411440521 JoshLee \**  *\* Date: 2023/9/22 \**  *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*  #include <stdio.h>  struct stint {  int f1, f2, f3; *// (total: 4 \* 3 = 12 bytes)*  };  struct stfloat {  float f4; *// (4 bytes) -> fill blank to 8 bytes*  double f5; *// (8 bytes)*  }; *// (total: 8 \* 2 = 16 bytes)*  struct sam {  char utype; *// (1 byte) -> fill blank to 4 bytes*  float f6, f7; *// (4 \* 2 = 8 bytes) -> fill blank to 12 bytes*  double f8; *// (8 bytes)*  union {  struct stint f11; *// (12 bytes)*  struct stfloat f12; *// (16 bytes)*  } f13; *// (total: 16 bytes)*  }; *// (struct total: 40 bytes)*  struct sam s[100]; *// (40 \* 100 = 4000 bytes)*  int main() {  *// s[27]Preparation*  printf("&s[27] = &s[0] + 27 \* sizeof(s[0])\n");  printf("&s[27] = %p + 27 \* 40 = %p\n", &s[0], (void \*)&s[0] + 27 \* sizeof(s[0]));  *// &s[27] = &s[0] + 27 \* 40*  printf("\n");  *// 1. s[27].utype*  printf("1. s[27].utype\n");  printf("&s[27].utype: %p\tsize: %zu byte\n", &s[27].utype, sizeof(s[27].utype));  printf("&s[27] = &s[27].utype\n");  printf("%p = %p\n", &s[27], &s[27].utype);  *// starting address of s[27] is the starting address of the first element s[27].utype*  printf("\n");    *// 2. s[27].f6*  printf("2. s[27].f6\n&s[27].f6: %p\tsize: %zu bytes\n", &s[27].f6, sizeof(s[27].f6));  printf("&s[27].f6 = &s[27].utype + 3(blank) + 1(starting point)\n");  printf("%p + %x + %x = %p\n", &s[27].utype, 3, 1, (void \*)&s[27].utype + (unsigned int)3 + (unsigned int)1);  *// &s[27].f6 = &s[27].utype + 3(blank) + 1;*  printf("\n");  *// s[18]Preparation*  printf("&s[18] = &s[0] + 18 \* sizeof(s[0])\n");  printf("&s[18] = %p + 18 \* 40 = %p\n", &s[0], (void \*)&s[0] + 18 \* sizeof(s[0]));  *// &s[18] = &s[0] + 18 \* 40*  printf("\n");    *// 3. s[18].f8*  printf("3. s[18].f8\n&s[18].f8: %p\tsize: %zu bytes\n", &s[18].f8, sizeof(s[18].f8));  printf("&s[18].f8 = &s[18].utype + 3(blank) + sizeof(float)\*2 + 4(blank) + 1(starting point)\n");  printf("%p + %x + %lx + %x + %x = %p\n", &s[18].utype, (unsigned int)3, sizeof(float)\*2, (unsigned int)4, (unsigned int)1, (void \*)&s[18].utype + (unsigned int)3 + sizeof(float)\*2 + (unsigned int)4 + (unsigned int)1);  *// &s[18].f8 = &s[18].utype + 3(blank) + 1 + sizeof(float)\*2 + 4(blank) + 1;*  printf("\n");  *// s[37]Preparation*  printf("&s[37] = &s[0] + 37 \* sizeof(s[0])\n");  printf("&s[37] = %p + 37 \* 40 = %p\n", &s[0], (void \*)&s[0] + 37 \* sizeof(s[0]));  *// &s[37] = &s[0] + 37 \* 40*  printf("\n");    *// 4. s[37].f13.f11.f3*  printf("4. s[37].f13.f11.f3\n&s[37].f13.f11.f3: %p\tsize: %zu bytes\n", &s[37].f13.f11.f3, sizeof(s[37].f13.f11.f3));  printf("&s[37].f13.f11.f3 = &s[37].utype + 3(blank) + sizeof(float)\*2 + 4(blank) + sizeof(double) + sizeof(int)\*2 + 1(starting point)\n");  printf("%p + %x + %lx + %x + %lx + %lx + %x = %p\n", &s[37].utype, (unsigned int)3, sizeof(float)\*2, (unsigned int)4, sizeof(double), sizeof(int)\*2, (unsigned int)1, (void \*)&s[37].utype + (unsigned int)3 + sizeof(float)\*2 + (unsigned int)4 + sizeof(double) + sizeof(int)\*2 + (unsigned int)1);  *// &s[37].f13.f11.f3 = &s[37].utype + 3(blank) + sizeof(float)\*2 + 4(blank) + sizeof(double) + sizeof(int)\*2 + 1(starting point);*  printf("\n");    *// 5. s[37].f13.f12.f5*  printf("5. s[37].f13.f12.f5\n&s[37].f13.f12.f5: %p\tsize: %zu bytes\n", &s[37].f13.f12.f5, sizeof(s[37].f13.f12.f5));  printf("&s[37].f13.f12.f5 = &s[37].utype + 3(blank) + sizeof(float)\*2 + 4(blank) + sizeof(double) + sizeof(float) + 4(blank) + 1(starting point)\n");  printf("%p + %x + %lx + %x + %lx + %lx + %x + %x = %p\n", &s[37].utype, (unsigned int)3, sizeof(float)\*2, (unsigned int)4, sizeof(double), sizeof(float), (unsigned int)4, (unsigned int)1, (void \*)&s[37].utype + (unsigned int)3 + sizeof(float)\*2 + (unsigned int)4 + sizeof(double) + sizeof(float) + (unsigned int)4 + (unsigned int)1);  *// &s[37].f13.f12.f5 = &s[37].utype + 3(blank) + sizeof(float)\*2 + 4(blank) + sizeof(double) + sizeof(float) + 4(blank) + 1(starting point);*  printf("\n");  return 0;  } |