得分

第三题 机器级编程（15 分，每空 1 分）

下面的 C 程序包含 `main()`, `caller()`, `callee()`三个函数。本题给出了该程序的部分 C 代码和 X86-64 汇编与机器代码。请分析给出的代码，补全空白处的内容，并回答问题。

注：汇编与机器码中的数字用 16 进制数填写

X86-64 汇编与机器代码：

答案填写处：

```
00000000004006cd <caller>:
  4006cd: 55                    push   %rbp
  4006ce: 48 89 e5              mov    %rsp,    %rbp
  4006d1: 48 83 ec 50           sub    $0x50,   %rsp
  4006d5: 48 89 7d b8           mov    %rdi,    -0x48(%rbp)
  4006d9: 64 48 8b 04 25 28 00  mov    %fs:0x28, %rax
  4006e0: 00 00
  4006e2: 48 89 45 f8           mov    %rax,    -0x8(%rbp)
  4006e6: 31 c0                 xor    %eax,    %eax
  4006e8: c6 45 d0 00           movb   $0x0,    -0x30(%rbp)
  4006ec: c6 45 e0 00           movb   $0x0,    (1)
  4006f0: 48 8b 45 b8           mov    (2),     %rax
  4006f4: 48 89 c7              mov    %rax,    %rdi
  4006f7:                       callq  400510 <strlen@plt>
  4006fc: 89 45 cc              mov    (3),     -0x34(%rbp)
  4006ff: 83 7d cc 0e           cmpl   $0xe,    -0x34(%rbp)
  400703: 7f (4)                jg     400752 <caller+0x85>
  400705: 83 7d cc 09           cmpl   $0x9,    -0x34(%rbp)
  400709:                       jg     400720 <caller+0x53>
  40070b: 48 8b 55 b8           mov    -0x48(%rbp), %rdx
  40070f: 48 8d 45 d0           lea    (5),     %rax
  400713: 48 89 d6              mov    %rdx,    %rsi
  400716: 48 89 c7              mov    %rax,    %rdi
  400719:                       callq  400500 <strcpy@plt>
  40071e:                       jmp    40073b <caller+0x6e>
  400720: 48 8b 45 b8           mov    -0x48(%rbp), %rax
  400724: 48 8d 50 0a           lea    0xa(%rax),  %rdx
  400728: 48 8d 45 d0           lea    -0x30(%rbp), %rax
  40072c: 48 83 c0 10           add    (6),     %rax
```

(1) -0x28(%rbp)

(2) -0x48(%rbp)

(3) %rax

(4) 4d

(5) -0x30(%rbp)

(6) 0x8

```
400730: 48 89 d6        mov     %rdx,       %rsi
400733: 48 89 c7        mov     %rax,       %rdi
400736:                 callq   400500 <strcpy@plt>
40073b: ff 75 e8        pushq   -0x18(%rbp)
40073e: ff 75 e0        pushq   -0x20(%rbp)
400741: ff 75 d8        pushq   -0x28(%rbp)
400744: ff 75 d0        pushq   -0x30(%rbp)
400747: e8 (7)          callq   400666 <callee>          (7) 1a ## ##
40074c: 48 83 c4 20     add     $0x20,      %rsp
400750:                 jmp     400753 <caller+0x86>
400752: 90              nop
400753: 48 8b 45 f8     mov     (8),        %rax         (8) -0x8(%rbp)
400757: 64 48 33 04 25 28 00   xor   %fs:0x28,  %rax
40075e: 00 00
400760:                 je      400767 <caller+0x9a>
400762:                 callq   400520 <__stack_chk_fail@plt>
400767: c9              leaveq  <=>  movq %rbp, %rsp
400768: c3              retq         popq %rbp
```

Call strcpy

push struct into stack

check canary

C 代码：

答案填写处：

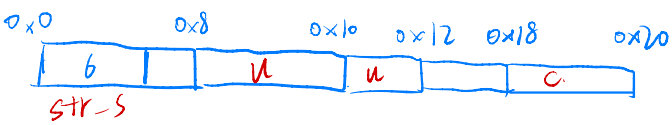
```
#include <stdio.h>
#include "string.h"
#define N    (9)
#define M    (10)

typedef union {char str_u[N]; long l;} union_e;
typedef struct {char str_s[M]; union_e u; long c;} struct_e;

void callee(struct_e s){
  char buf[M+N];
  strcpy(buf, s.str_s);
  strcat(buf, s.u.str_u);
  printf("%s \n",buf);
}
```

(9) 10

(10) 5



```
void caller(char *str){
  struct_e s;
  s.str_s[0]='\0';
  s.u.str_u[0]='\0';
  int len = strlen(str);
  if(len>=  M+N)
    (11) ;
  else if(len<N){
    strcpy(s.str_s, (12) );
  }
  else{
    strcpy(s.u.str_u, (13) );
  }
  callee(s);
}

int main(int argc, char *argv[]){
  caller("0123456789abcd");
  return 0;
}
```

len = 14

(11) return;

(12) str

(13) str+10 / str+M

%rdi %rsi

%rdi %rsi

"abcd"

caller 函数中，变量 s 所占的内存空间为: (14) 0x20 bytes

该程序运行后，printf 函数是否有输出？输出结果为: (15) abcd

答案：

机器级编程（15 分，每空 1 分）

下面的 C 程序包含 main(), caller(), callee()三个函数。本题给出了该程序的部分 C 代码和 X86-64 汇编与机器代码。请分析给出的代码，补全空白处的内容，并回答问题。

注：汇编与机器码中的数字用 16 进制数填写

X86-64 汇编与机器代码： 答案填写处：

```
00000000004006cd <caller>:
   4006cd: 55                      push    %rbp
   4006ce: 48 89 e5                mov     %rsp,    %rbp
   4006d1: 48 83 ec 50             sub     $0x50,   %rsp
   4006d5: 48 89 7d b8             mov     %rdi,    -0x48(%rbp)
   4006d9: 64 48 8b 04 25 28 00    mov     %fs:0x28,  %rax
```

```
4006e0: 00 00
4006e2: 48 89 45 f8     mov   %rax,      -0x8(%rbp)
4006e6: 31 c0           xor   %eax,      %eax
4006e8: c6 45 d0 00     movb  $0x0,      -0x30(%rbp)
4006ec: c6 45 e0 00     movb  $0x0,      (1)          (1) -0x28(%rbp)
4006f0: 48 8b 45 b8     mov   (2),       %rax         (2) -0x48(%rbp)
4006f4: 48 89 c7        mov   %rax,      %rdi
4006f7:                 callq 400510 <strlen@plt>
4006fc:  89 45 cc       mov   (3),       -0x34(%rbp)  (3) %eax
4006ff:  83 7d cc 0e    cmpl  $0xe,      -0x34(%rbp)
400703: 7f (4)          jg    400752 <caller+0x85>   (4) 4d
400705: 83 7d cc 09     cmpl  $0x9,      -0x34(%rbp)
400709:                 jg    400720 <caller+0x53>
40070b: 48 8b 55 b8     mov   -0x48(%rbp), %rdx
40070f: 48 8d 45 d0     lea   (5),       %rax         (5) -0x30(%rbp)
400713: 48 89 d6        mov   %rdx,      %rsi
400716: 48 89 c7        mov   %rax,      %rdi
400719:                 callq 400500 <strcpy@plt>
40071e:                 jmp   40073b <caller+0x6e>
400720: 48 8b 45 b8     mov   -0x48(%rbp), %rax
400724: 48 8d 50 0a     lea   0xa(%rax),  %rdx
400728: 48 8d 45 d0     lea   -0x30(%rbp), %rax
40072c: 48 83 c0 10     add   (6),       %rax         (6) 0x8
400730: 48 89 d6        mov   %rdx,      %rsi
400733: 48 89 c7        mov   %rax,      %rdi
400736:                 callq 400500 <strcpy@plt>
40073b: ff 75 e8        pushq -0x18(%rbp)
40073e: ff 75 e0        pushq -0x20(%rbp)
400741: ff 75 d8        pushq -0x28(%rbp)
400744: ff 75 d0        pushq -0x30(%rbp)
400747: e8 (7)          callq 400666 <callee>      (7) 1a ff ff ff
40074c: 48 83 c4 20     add   $0x20,     %rsp
400750:                 jmp   400753 <caller+0x86>
400752: 90              nop
400753: 48 8b 45 f8     mov   (8),       %rax         (8) -0x8(%rbp)
400757: 64 48 33 04 25 28 00  xor   %fs:0x28,  %rax
40075e: 00 00
```

```
400760:                    je      400767 <caller+0x9a>
400762:                    callq   400520 <__stack_chk_fail@plt>
400767: c9                 leaveq
400768: c3                 retq
```

C 代码：　　答案填写处：

```
#include <stdio.h>
#include "string.h"
#define N    (9)
#define M    (10)

typedef union {char str_u[N]; long l;} union_e;
typedef struct {char str_s[M]; union_e u; long c;} struct_e;

void callee(struct_e s){
  char buf[M+N];
  strcpy(buf, s.str_s);
  strcat(buf, s.u.str_u);
  printf("%s \n",buf);
}

void caller(char *str){
  struct_e s;
  s.str_s[0]='\0';
  s.u.str_u[0]='\0';
  int len = strlen(str);
  if(len>=   M+N)
    (11) ;
  else if(len<N){
    strcpy(s.str_s, (12) );
  }
  else{
    strcpy(s.u.str_u, (13) );
  }
  callee(s);
```

(9)　10

(10)　5

(11)　return

(12)　str

(13)　str+M

```
}

int main(int argc, char *argv[]){
  caller("0123456789abcd");
  return 0;
}
```

caller 函数中，变量 s 所占的内存空间为: (14) 32 字节

该程序运行后，printf 函数是否有输出？输出结果为: (15) abcd