C Programming
Day14.B

2017.11.01

strtod(), strcat

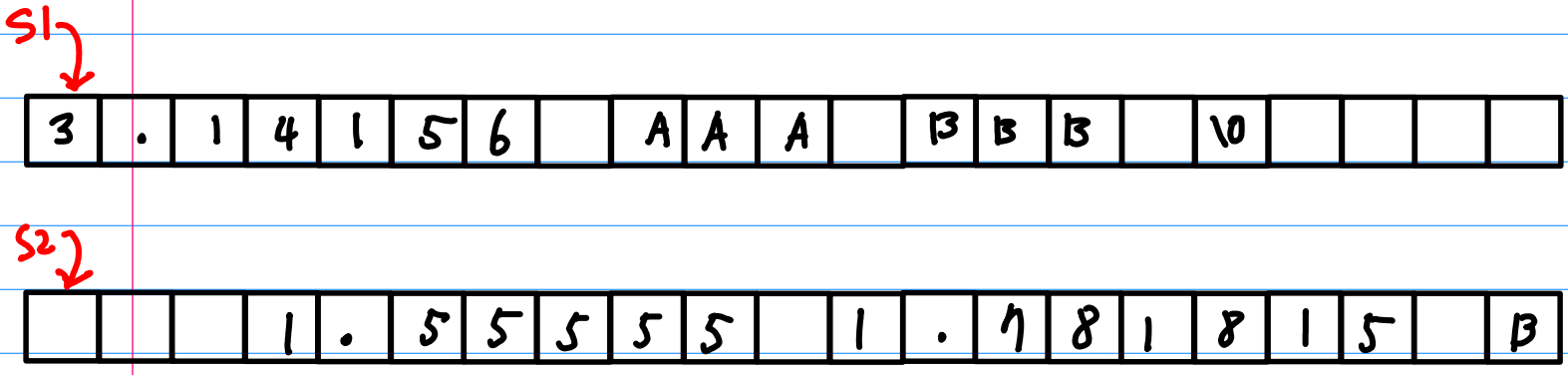
```c
#include <stdio.h>
#include <stdlib.h>

int main(void) {
  char *S1 = "3.14156 AAA BBB ";
  char *S2 = "   1.55555 1.781815 BBB ";
  char *N1;
  char *N2;
  double x1, x2;

  printf("&S1= %p S1= %p *S1= %c\n", &S1, S1, *S1);
  printf("&S2= %p S2= %p *S2= %c\n", &S2, S2, *S2);

  printf("S1 = \"%s\" \n", S1);
  printf("S2 = \"%s\" \n", S2);

  x1 = strtod(S1, &N1);
  printf("x1= %lf \n", x1);

  x2 = strtod(S2, &N2);
  printf("x2= %lf \n", x2);

  printf("N1 = \"%s\" \n", N1);
  printf("N2 = \"%s\" \n", N2);

  x1 = strtod(N1, &N1);
  printf("x1= %lf \n", x1);

  x2 = strtod(N2, &N2);
  printf("x2= %lf \n", x2);

  printf("N1 = \"%s\" \n", N1);
  printf("N2 = \"%s\" \n", N2);

}
```

```
&S1= 0x7ffe3800a428 S1= 0x400854 *S1= 3
&S2= 0x7ffe3800a430 S2= 0x400865 *S2=
S1 = "3.14156 AAA BBB "
S2 = "   1.55555 1.781815 BBB "
x1= 3.141560
x2= 1.555550
N1 = " AAA BBB "
N2 = " 1.781815 BBB "
x1= 0.000000
x2= 1.781815
N1 = " AAA BBB "
N2 = " BBB "
```

unsuccesful

```
x1 = strtod(S1, &N1);
printf("x1= %lf \n", x1);

x2 = strtod(S2, &N2);
printf("x2= %lf \n", x2);
```

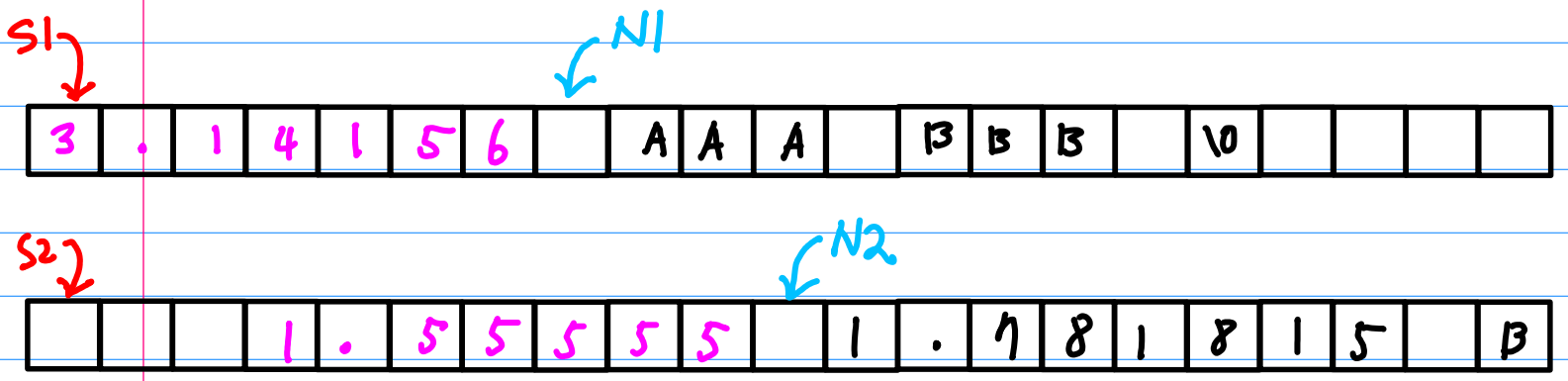
```
S1 = "3.14156 AAA BBB "
S2 = "    1.55555 1.781815 BBB "
```

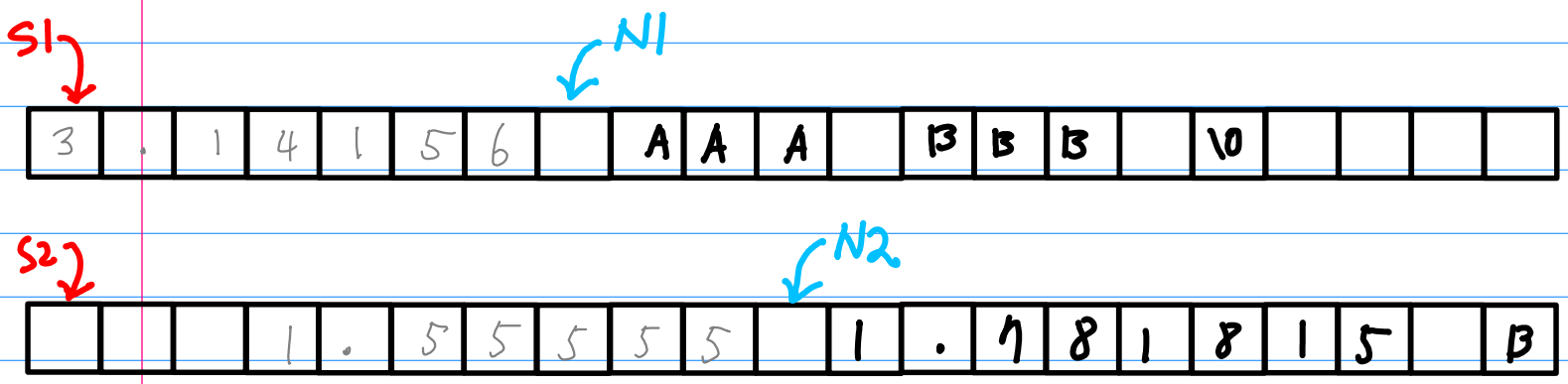


output

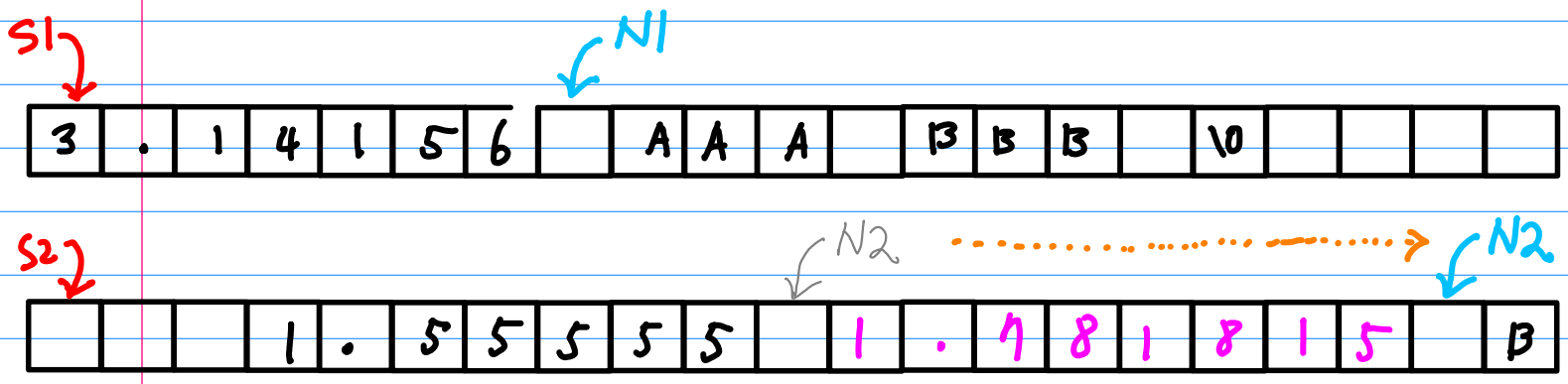
```
x1 = strtod(S1, &N1);
printf("x1= %lf \n", x1);

x2 = strtod(S2, &N2);
printf("x2= %lf \n", x2);
```

output






```
x1 = strtod(N1, &N1);
printf("x1= %lf \n", x1);

x2 = strtod(N2, &N2);
printf("x2= %lf \n", x2);
```



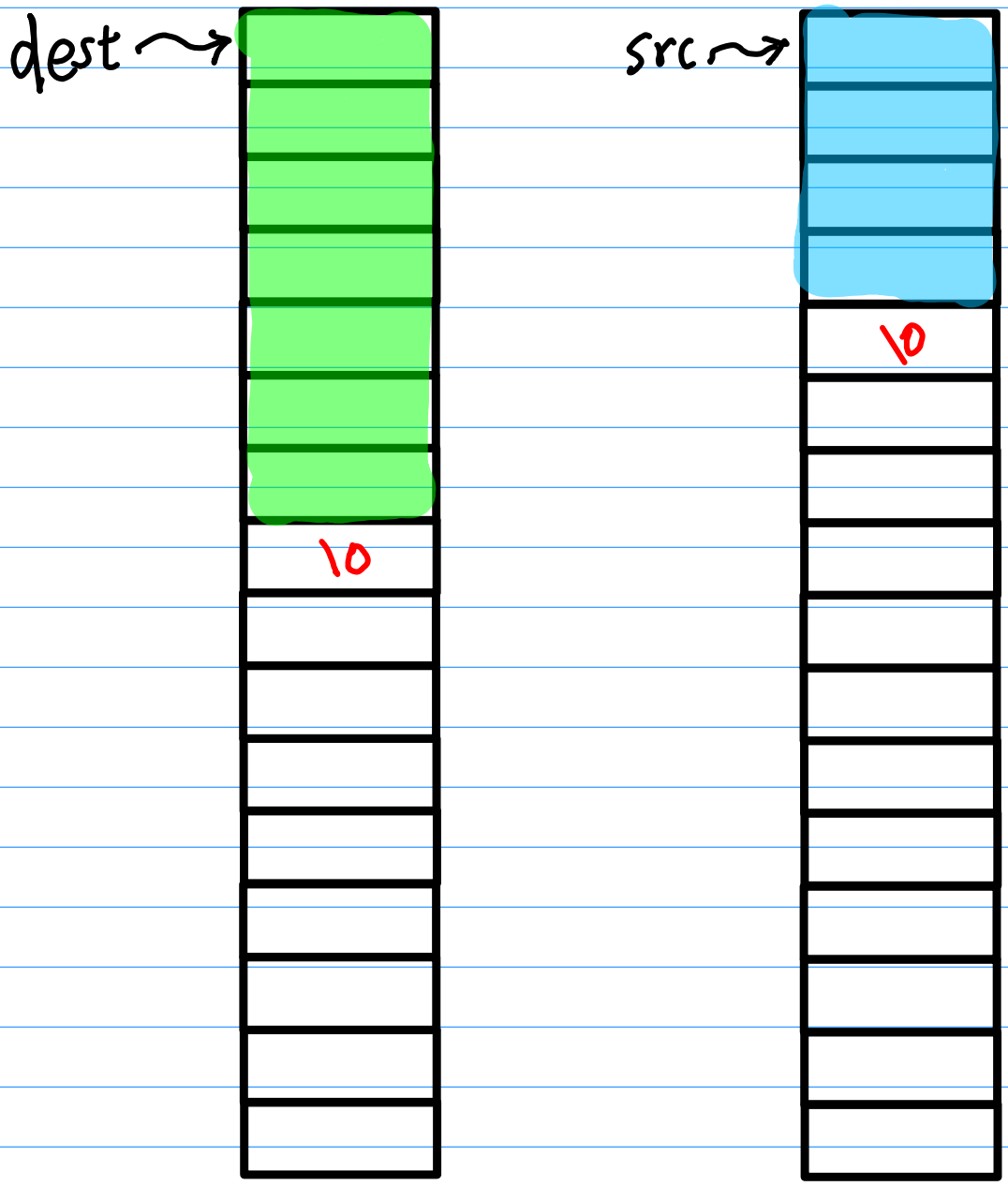
```c
#include <stdio.h>
#include <stdlib.h>


int main(void) {
  char S[] = "111.1 222.2 333.3 444.4 555.5 AAA BBB";
  char *p, *q, *old_p;
  double x;


  p = old_p = S; q = NULL;

  while (old_p != q) {
    x = strtod(p, &q);
    old_p = p;
    p = q;
    printf(" %f\n", x);
  }

}
```

```
111.100000
222.200000
333.300000
444.400000
555.500000
0.000000
```

```c
#include <stdio.h>
#include <stdlib.h>


int main(void) {
  char S[] = "111.1 222.2 333.3 444.4 555.5 AAA BBB";
  char *p, *q, *old_p;
  double x;

  for (int i=0; i<sizeof(S); ++i)
    printf("S[%2d]= %c : %p \n", i, S[i], S+i);

  p = old_p = S; q = NULL;

  while (old_p != q) {
    printf("%p : ", p);
    x = strtod(p, &q);
    old_p = p;
    p = q;
    printf(" %f\n", x);
  }

}
```

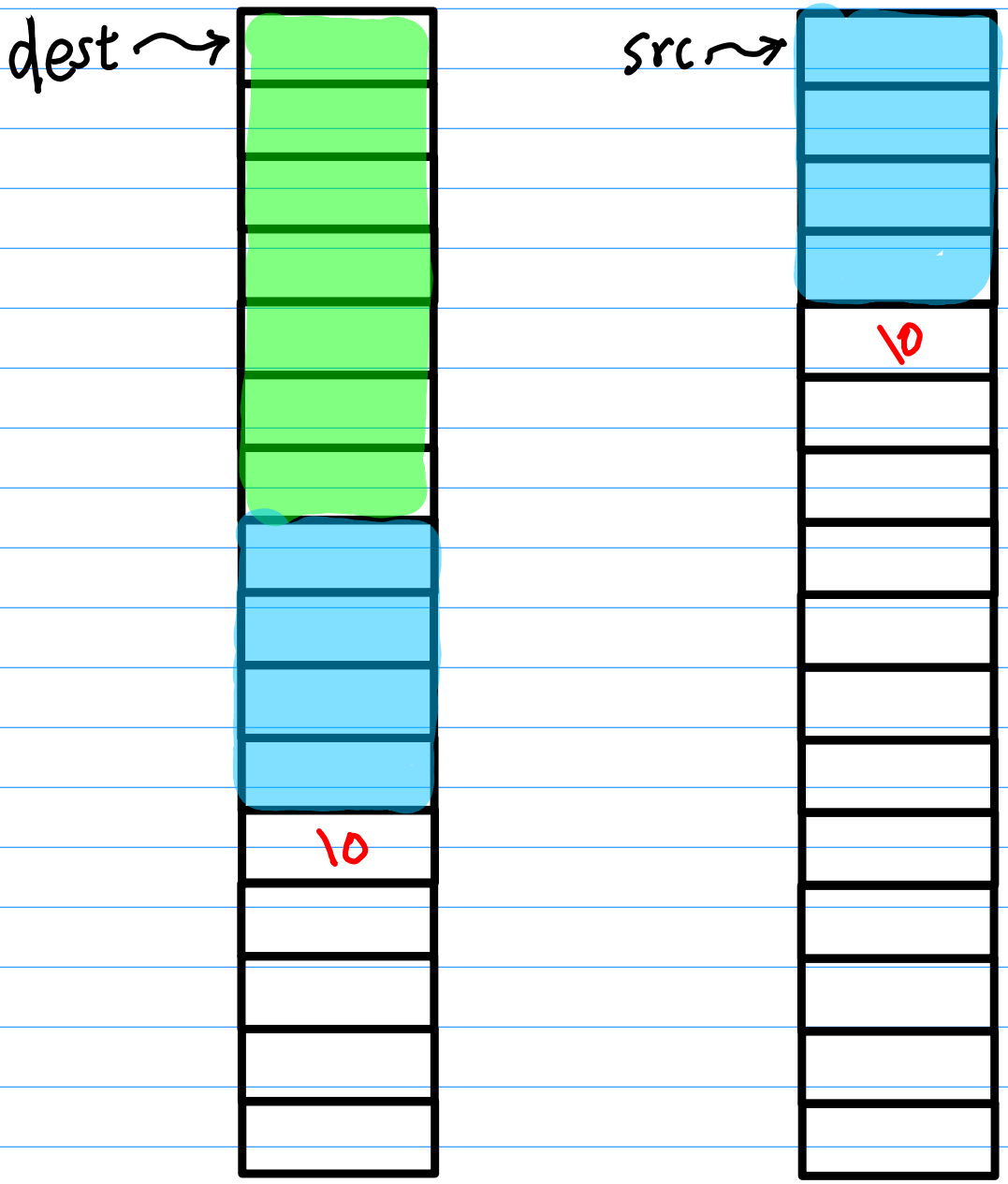
```
S[ 0]= 1 : 0x7ffc58f754b0
S[ 1]= 1 : 0x7ffc58f754b1
S[ 2]= 1 : 0x7ffc58f754b2
S[ 3]= . : 0x7ffc58f754b3
S[ 4]= 1 : 0x7ffc58f754b4
S[ 5]=   : 0x7ffc58f754b5
S[ 6]= 2 : 0x7ffc58f754b6
S[ 7]= 2 : 0x7ffc58f754b7
S[ 8]= 2 : 0x7ffc58f754b8
S[ 9]= . : 0x7ffc58f754b9
S[10]= 2 : 0x7ffc58f754ba
S[11]=   : 0x7ffc58f754bb
S[12]= 3 : 0x7ffc58f754bc
S[13]= 3 : 0x7ffc58f754bd
S[14]= 3 : 0x7ffc58f754be
S[15]= . : 0x7ffc58f754bf
S[16]= 3 : 0x7ffc58f754c0
S[17]=   : 0x7ffc58f754c1
S[18]= 4 : 0x7ffc58f754c2
S[19]= 4 : 0x7ffc58f754c3
S[20]= 4 : 0x7ffc58f754c4
S[21]= . : 0x7ffc58f754c5
S[22]= 4 : 0x7ffc58f754c6
S[23]=   : 0x7ffc58f754c7
S[24]= 5 : 0x7ffc58f754c8
S[25]= 5 : 0x7ffc58f754c9
S[26]= 5 : 0x7ffc58f754ca
S[27]= . : 0x7ffc58f754cb
S[28]= 5 : 0x7ffc58f754cc
S[29]=   : 0x7ffc58f754cd
S[30]= A : 0x7ffc58f754ce
S[31]= A : 0x7ffc58f754cf
S[32]= A : 0x7ffc58f754d0
S[33]=   : 0x7ffc58f754d1
S[34]= B : 0x7ffc58f754d2
S[35]= B : 0x7ffc58f754d3
S[36]= B : 0x7ffc58f754d4
S[37]=  : 0x7ffc58f754d5
0x7ffc58f754b0 :  111.100000
0x7ffc58f754b5 :  222.200000
0x7ffc58f754bb :  333.300000
0x7ffc58f754c1 :  444.400000
0x7ffc58f754c7 :  555.500000
0x7ffc58f754cd :  0.000000
```

char *strcat(char *dest, const char *src);



```
char *strcat(char *dest, const char *src);
```

return dest



```c
#include <stdio.h>
#include <string.h>

int main(void) {
  char S1[] = "Happy ";
  char S2[] = "New Year ";

  printf("S1= %s \n", S1);
  printf("S2= %s \n", S2);

  printf("strcat(S1,S2)= %s \n", strcat(S1, S2));

  printf("S1= %s \n", S1);
  printf("S2= %s \n", S2);

  strcat(S1, S2);

  printf("S1= %s \n", S1);
  printf("S2= %s \n", S2);
}
```

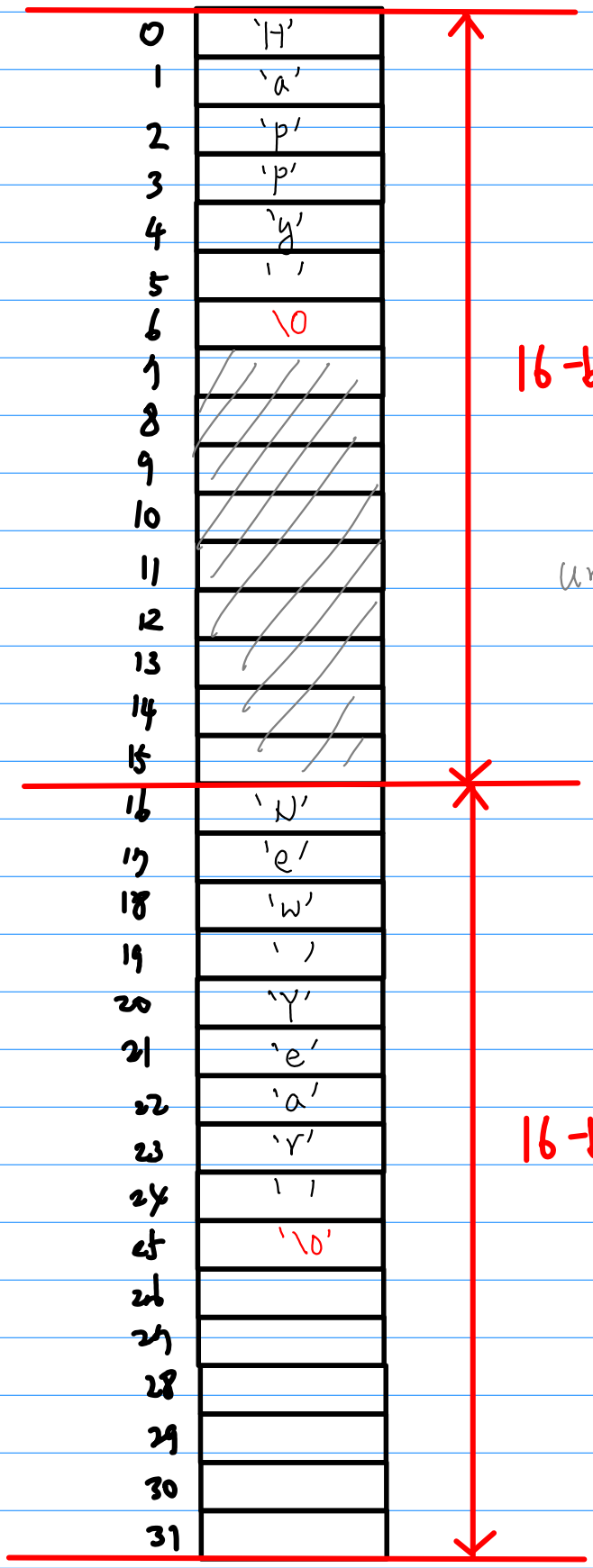
char *strcat(char *dest, const char *src);

```
S1= Happy
S2= New Year
strcat(S1,S2)= Happy New Year
S1= Happy New Year
S2= New Year
S1= Happy New Year New Year
S2= ew Year
```

Why ?

```c
#include <stdio.h>
#include <string.h>

void prstring(char *S, char *N) {
  int   i;
  char *p;

  puts("------------------------------");
  p = S; i = 0;
  while (1) {
    printf("%s+%2d= %p : ", N, i, S+i);
    printf("%1c %2x\n", *(S+i), *(S+i));
    if (*p) { p++; i++; }
    else return;
  }
}

int main(void) {
  char S1[] = "Happy ";
  char S2[] = "New Year ";

  printf("S1= %s \n", S1);
  printf("S2= %s \n", S2);

  prstring(S1, "S1");
  prstring(S2, "S2");

  puts("\n.............................");
  printf("strcat(S1,S2)= %s \n", strcat(S1, S2));
  puts(".............................\n");

  printf("S1= %s \n", S1);
  printf("S2= %s \n", S2);

  prstring(S1, "S1");
  prstring(S2, "S2");

  puts("\n.............................");
  printf("strcat(S1,S2)\n");
  strcat(S1, S2);
  puts(".............................\n");

  printf("S1= %s \n", S1);
  printf("S2= %s \n", S2);

  prstring(S1, "S1");
  prstring(S2, "S2");

}
```

```
char S1[] = "Happy ";
char S2[] = "New Year "
```



unused

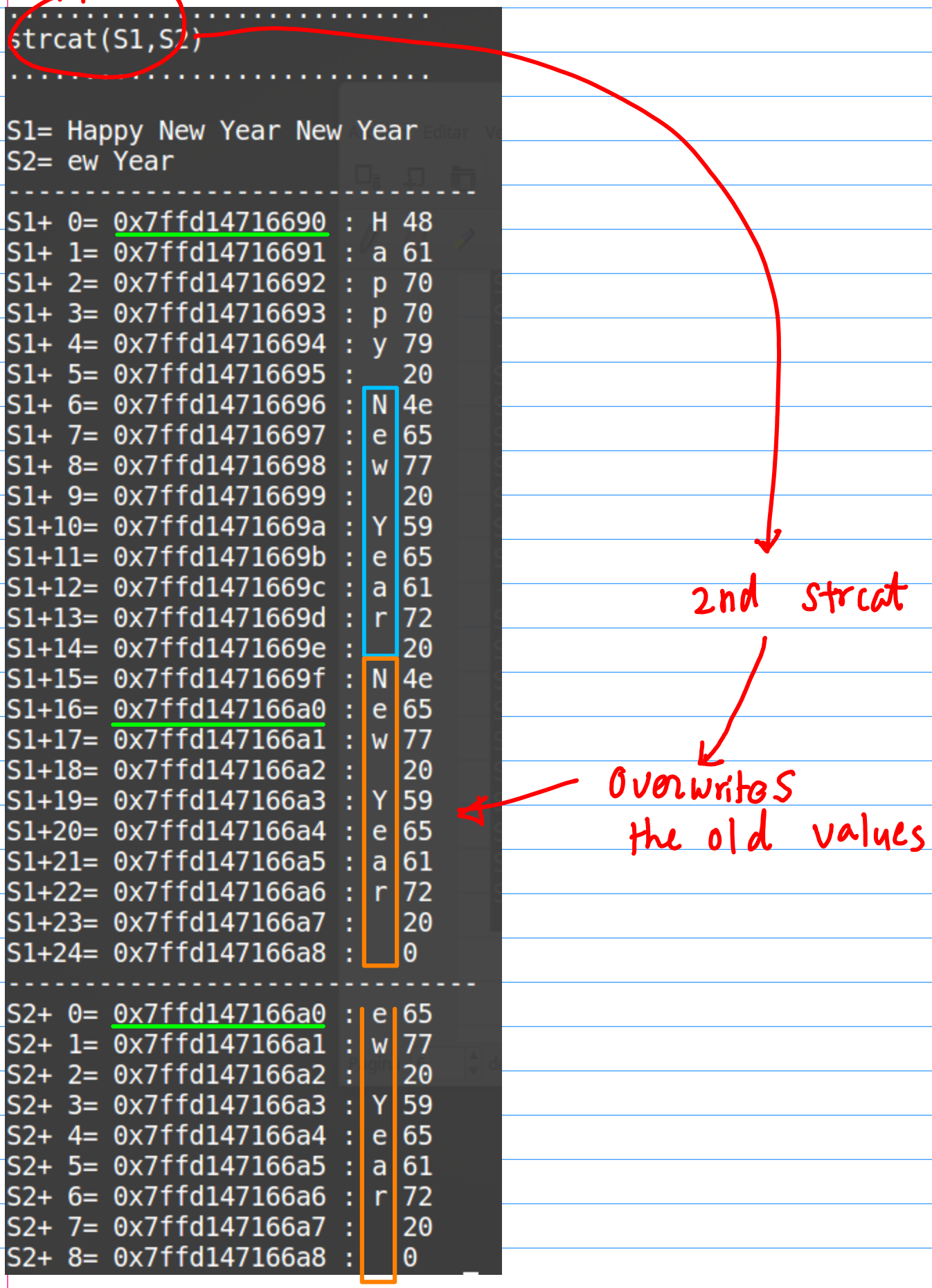
memory alignment

next 16-byte boundary

performance reaso

```
S1= Happy
S2= New Year
------------------------------
S1+ 0= 0x7ffd14716690 : H 48
S1+ 1= 0x7ffd14716691 : a 61
S1+ 2= 0x7ffd14716692 : p 70
S1+ 3= 0x7ffd14716693 : p 70
S1+ 4= 0x7ffd14716694 : y 79
S1+ 5= 0x7ffd14716695 :   20
S1+ 6= 0x7ffd14716696 :   0
------------------------------
S2+ 0= 0x7ffd147166a0 : N 4e
S2+ 1= 0x7ffd147166a1 : e 65
S2+ 2= 0x7ffd147166a2 : w 77
S2+ 3= 0x7ffd147166a3 :   20
S2+ 4= 0x7ffd147166a4 : Y 59
S2+ 5= 0x7ffd147166a5 : e 65
S2+ 6= 0x7ffd147166a6 : a 61
S2+ 7= 0x7ffd147166a7 : r 72
S2+ 8= 0x7ffd147166a8 :   20
S2+ 9= 0x7ffd147166a9 :   0
```

```
..............................
strcat(S1,S2)= Happy New Year
..............................

S1= Happy New Year
S2= New Year
------------------------------
S1+ 0= 0x7ffd14716690 : H 48
S1+ 1= 0x7ffd14716691 : a 61
S1+ 2= 0x7ffd14716692 : p 70
S1+ 3= 0x7ffd14716693 : p 70
S1+ 4= 0x7ffd14716694 : y 79
S1+ 5= 0x7ffd14716695 :   20
S1+ 6= 0x7ffd14716696 : N 4e
S1+ 7= 0x7ffd14716697 : e 65
S1+ 8= 0x7ffd14716698 : w 77
S1+ 9= 0x7ffd14716699 :   20
S1+10= 0x7ffd1471669a : Y 59
S1+11= 0x7ffd1471669b : e 65
S1+12= 0x7ffd1471669c : a 61
S1+13= 0x7ffd1471669d : r 72
S1+14= 0x7ffd1471669e :   20
S1+15= 0x7ffd1471669f :   0
------------------------------
S2+ 0= 0x7ffd147166a0 : N 4e
S2+ 1= 0x7ffd147166a1 : e 65
S2+ 2= 0x7ffd147166a2 : w 77
S2+ 3= 0x7ffd147166a3 :   20
S2+ 4= 0x7ffd147166a4 : Y 59
S2+ 5= 0x7ffd147166a5 : e 65
S2+ 6= 0x7ffd147166a6 : a 61
S2+ 7= 0x7ffd147166a7 : r 72
S2+ 8= 0x7ffd147166a8 :   20
S2+ 9= 0x7ffd147166a9 :   0
```


.....
strcat(S1,S2)
.....
S1= Happy New Year New Year
S2= ew Year
S1+ 0= 0x7ffd14716690 : H 48
S1+ 1= 0x7ffd14716691 : a 61
S1+ 2= 0x7ffd14716692 : p 70
S1+ 3= 0x7ffd14716693 : p 70
S1+ 4= 0x7ffd14716694 : y 79
S1+ 5= 0x7ffd14716695 : 20
S1+ 6= 0x7ffd14716696 : N 4e
S1+ 7= 0x7ffd14716697 : e 65
S1+ 8= 0x7ffd14716698 : w 77
S1+ 9= 0x7ffd14716699 : 20
S1+10= 0x7ffd1471669a : Y 59
S1+11= 0x7ffd1471669b : e 65
S1+12= 0x7ffd1471669c : a 61
S1+13= 0x7ffd1471669d : r 72
S1+14= 0x7ffd1471669e : 20
S1+15= 0x7ffd1471669f : N 4e
S1+16= 0x7ffd147166a0 : e 65
S1+17= 0x7ffd147166a1 : w 77
S1+18= 0x7ffd147166a2 : 20
S1+19= 0x7ffd147166a3 : Y 59
S1+20= 0x7ffd147166a4 : e 65
S1+21= 0x7ffd147166a5 : a 61
S1+22= 0x7ffd147166a6 : r 72
S1+23= 0x7ffd147166a7 : 20
S1+24= 0x7ffd147166a8 : 0
S2+ 0= 0x7ffd147166a0 : e 65
S2+ 1= 0x7ffd147166a1 : w 77
S2+ 2= 0x7ffd147166a2 : 20
S2+ 3= 0x7ffd147166a3 : Y 59
S2+ 4= 0x7ffd147166a4 : e 65
S2+ 5= 0x7ffd147166a5 : a 61
S2+ 6= 0x7ffd147166a6 : r 72
S2+ 7= 0x7ffd147166a7 : 20
S2+ 8= 0x7ffd147166a8 : 0
2nd strcat
overwrites the old values