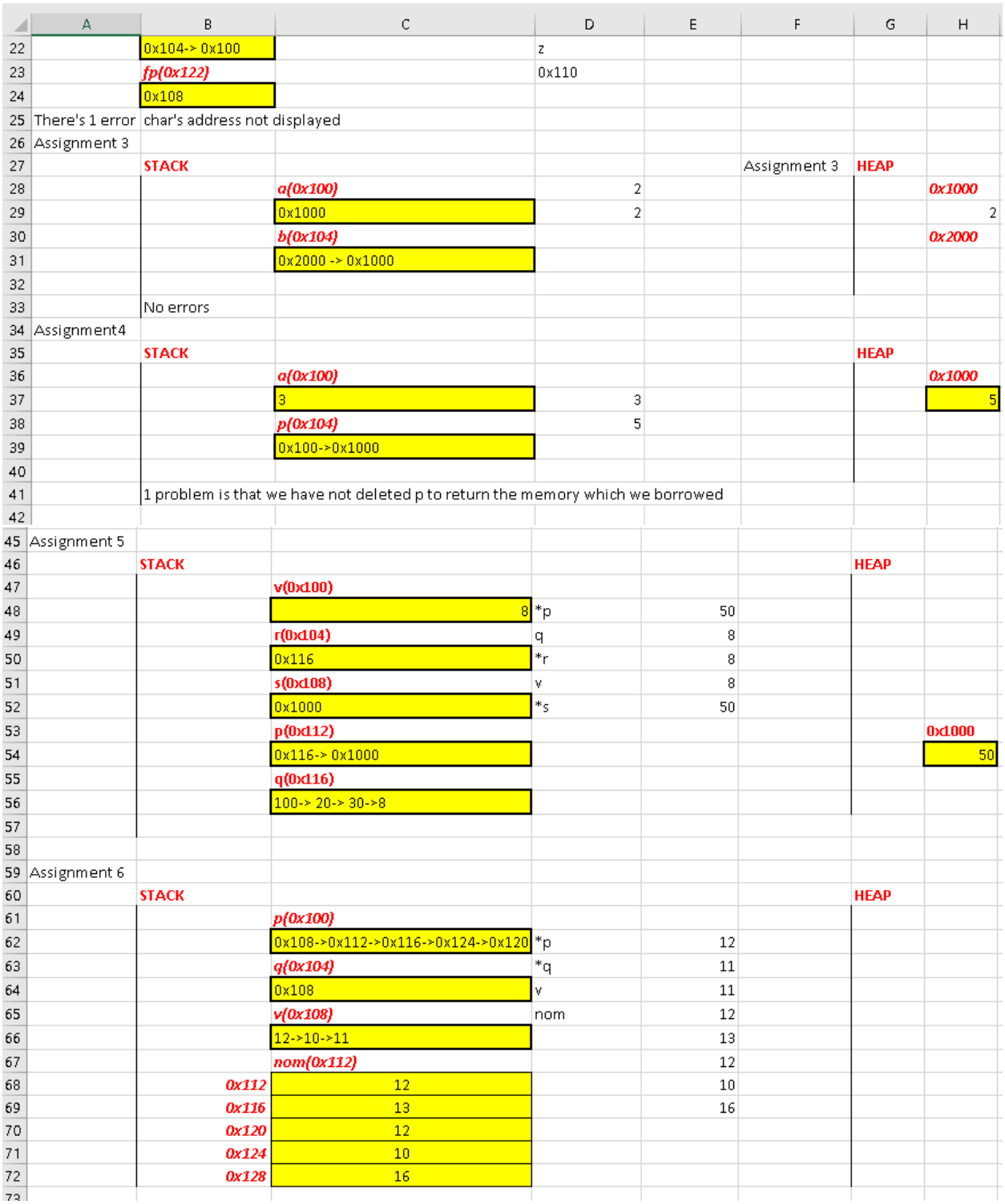
	A	B	C	D
1	Assignment 1			
2		***a (0x100)***		0x100
3		3		3
4		***b(0x104)***		0x104
5		0x100		3
6				0x100
7	No errors			
8	Assignment 2			
9		***x (0x100)***		100
10		100		1,2
11		***z(0x104)***		400
12		50->100->200->400		0x104
13		***y(0x108)***		400
14		20,0->1,2		0x114
15		***ch(0x109)***		0x100
16		Z		100
17		***chp(0x110)***		0x118
18		0x109		0x108
19		***ip1(0x114)***		1,2
20		0x100->0x104		0x122
21		***ip2(0x118)***		Error
22		0x104-> 0x100		z
23		***fp(0x122)***		0x110
24		0x108		
25	There's 1 error	char's address not displayed		



	A	B	C	D	E	F	G	H
22		0x104-> 0x100		z				
23		*fp(0x122)*		0x110				
24		0x108						
25	There's 1 error	char's address not displayed						
26	Assignment 3							
27		STACK				Assignment 3	HEAP	
28			*a(0x100)*	2				*0x1000*
29			0x1000	2				2
30			*b(0x104)*					*0x2000*
31			0x2000 -> 0x1000					
32								
33		No errors						
34	Assignment4							
35		STACK					HEAP	
36			*a(0x100)*					*0x1000*
37			3	3				5
38			*p(0x104)*	5				
39			0x100->0x1000					
40								
41		1 problem is that we have not deleted p to return the memory which we borrowed						
42								
45	Assignment 5							
46		STACK					HEAP	
47			v(0x100)					
48			8	*p	50			
49			r(0x104)	q	8			
50			0x116	*r	8			
51			s(0x108)	v	8			
52			0x1000	*s	50			
53			p(0x112)					0x1000
54			0x116-> 0x1000					50
55			q(0x116)					
56			100-> 20-> 30->8					
57								
58								
59	Assignment 6							
60		STACK					HEAP	
61			*p(0x100)*					
62			0x108->0x112->0x116->0x124->0x120	*p	12			
63			*q(0x104)*	*q	11			
64			0x108	v	11			
65			*v(0x108)*	nom	12			
66			12->10->11		13			
67			*nom(0x112)*		12			
68		*0x112*	12		10			
69		*0x116*	13		16			
70		*0x120*	12					
71		*0x124*	10					
72		*0x128*	16					

Assignment 7:

B. Error: suspicious pointer conversation

Because we haven't assigned pointer x for any address, so we can't used *x = 100 to change the value of the address to 100.

Assignment 9:(8)

D.ce

*s = "peace", when we print s++ +3 means we print s from the position 3th which is "ce"

Assignment 13: (9)

a)*

Assignment 14: (10)

a)x is a pointer to a string, y is a string

Assignment 15: (11)

d) point to a tye

Because a,b,c are correct.

Assignment 16: (12)

c) int i; double* dp = &I;

pointer dp and i must be the same type.

Assignment 17: (13)

b) p now points to b

Assignment 19: (14)

a)ABCDEFGHIJ

as for ASCII, *(arr + 0)=arr[0] = 65 which is A,*(arr + 1)= arr[1] = 66 which is B and so on.

Assignment 20: (15)

a)fg

ptr = Str = “abcdefg” so when ptr += 5, it will cout from the position 5^{th} of ptr.

Assignment 21: (16)

D.All of them

Assignment 24: (17)

C. The new operator

Assignment 25: (18)

B. Indirection

Assignment 27: (19)

A.Pointer contains an address of a variable

Assignment 28: (20)

C.3

0,NULL,address

Assignment 29: (21)

C.Address operator

Assignment 30: (22)

c.129,a

cho += a means cho += 32 means ch =’A’ = 65 +32 = 97 = ‘a’ (as cho is a reference variable to ch)

*ptr += ch means a += ‘a’ = 97 means a = 97 + 32 =129 (as *ptr returns the value of a)

Assignment 31: (23)

d.Compile error

ptr has been defined as a constant pointer so the value of it(i’s address) and I’s value can’t be changed. Therefore (*ptr)++ and ptr = &j made it error.

Assignment 33: (24)

c.14

*arr + 10 = arr[0] + 10 = 4 + 10 =14

Assignment 34: (25)

c. compile error

ra declared as reference but not initiallized

Assignment 35: (26)

2 will be the output because ptr = a means ptr = a[0] so *(ptr + 1) = a[0+1] = a[1] =2.

Assignment 36: (27)

15 will be the output because *ptr = &a so *ptr = *ptr * 3 = a *3 = 5*3 = 15

Assignment 37: (28)

222 will be the output because (I * *j * I + *j) = 6*6*6 +6 = 222 (* j = &I = 6).

Assignment 38: (29)

X = 20, &x =500, y = &x = 500;

Z = y = 500, *y++ = *(y+1) = 504 (next address to x's address) ;

*z++ = *(z+1) = 504; x++ = 21.

So **the output will be: 21, 504, 504**

Assignment 41: (30)

b) Runtime error

because ptr is a wild pointer and may point anywhere in memory so we can't *ptr = 5.