Solution Q1

CSC209H Worksheet: Stacks and Heaps

1. Trace the memory usage for the program below. We have set up both stack frames for you, and the location of the heap.

		Section	Address	Value	Label
		Heap	0x23¢		
			0x240		
			0x244		
			0x248	er en	
de <stdio.h></stdio.h>					
carrayl(int a, int b, int c) { : arr[3]; :[0] = a;		stack frame for mkaxray1	0x454		
	original,		0x458		
: *p = arr;		Hace	0x45c	0×464/	TP
urn p;			0x460		
n() {		a	0x464	20	T) a[
*ptr = mkarrayi(10, 20, 30); ler_function();			0x46c 0x470	30	
intf("%d %d %d\n", ptr[0], ptr[1 curn 0;], ptr[2]);		0x474	30	J C
			0x478	20	d JE
			0x47c	1_10] à
	a Pinnini Suderole	stack frame	0x480		
		for main	0x484		
			0x488	- A 111	
			0x48c	0×464	

- 2. The program in part 1 will not work correctly. Notice the call to other_function. Explain to your partner why the program doesn't work. Fix the mkarray1 function, and trace it again.
- 3. Once you've fixed the code, add a statement to your program to deallocate the memory on the heap as soon as possible.

Solution Q2+Q3

Section

Address

Value

answer shown

10

Label

CSC209H Worksheet: Stacks and Heaps

1. Trace the memory usage for the program below. We have set up both stack frames for you, and the location of the heap.

	Heap	0x23c	1 10 1	
		0x240	20	
		0x244	<u>L30</u>	
		0x248		ans an energia de la come O la la come de la come de Al la come de la come
#include <stdio.h></stdio.h>				
int *mkarrayi(int a, int b, int c) { int arr[3]; inf k arr = malloc arr[0] = a; inf k arr = malloc int *p = arr; return p; return arr; }	stack frame for mkarra) (A)	0x454 0x458 0x45c 0x460		when mkarrow returns.
		0x464		
int main() {		0x46c	0x 23 c	7 arr
<pre>int *ptr = mkarray1(10, 20, 30); other_function();</pre>		0x470 L		ゴ フ30
printf("%d %d %d\n", ptr[0], ptr[1], ptr[2]); return 0,\footnote{\sqrt{7}}		0x47#	20	$Z_{ m so}$
		0x478 L	2.0	$\exists $
		0x47c \		
	stack frame for main	0x480		
		0x484		
		0x488	0 x 23 (alptr
		0x48c	U X &O (

2. The program in part 1 will not work correctly. Notice the call to other_function. Explain to your partner why

3. Once you've fixed the code, add a statement to your program to deallocate the memory on the heap as soon as

the program doesn't work. Fix the mkarray1 function, and trace it again.

free (ptr);

solution for Q4

CSC209H Worksheet: Stacks and Heaps

4. Trace the memory usage for the program below. We have set up the stack frame for you, and the location of the heap. Solution Shown.

	Section	Address	Value	Label
include <stdio.h> include <stdlib.h></stdlib.h></stdio.h>	Неар	0x23c	3	
* Build an array in dynamic memory to hold		0x228	16	
multiples of x from x to x*x. Return a pointer to this array.		0 x 22c	19	
		0x230		
<pre>it *multiples(int x) { int *a = malloc(sizeof(int) * x);</pre>				ios del Especiela D'Especiela especie
for (int i=0; i < x; i++) { a[i] = (i+1) * x;		0x234		
}		0x238		
return a;		0x23¢		
	a i se se redon establica Constante no de se cipar	0x240	antenis in tarring in Substitution (Silandari	
nt main() { int *ptr;		0x244		
int size = 3;				
	stack frame for multiples	0x470	DX23	J. C.
<pre>ptr = multiples(size);</pre>	ang sa Shi sa sa sa sa sa	0±470	Ta 03	Ji!
		0x474	12x23 C	<u>`</u>
<pre>for (int i=0; i<size; i++)="" pre="" printf("%d\t",="" ptr[i]);<="" {=""></size;></pre>		0x478	3	J.Z.
<pre>} printf("\n");</pre>	stack frame	0x47c		
	for main	0x480		
		0x484	T 3] size
return 0;				
		0x488	10x23c	1 ptr
		0x48c	1	_1 T

- 5. Change the main function so that it calls multiples and prints the array in a loop with sizes of 3, 4, and 5. Besides the changes described, do not make any other changes or additions to the code.
- 6. Trace the memory usage of your changed program. Explain the problem to your partner and then fix it by adding calls to deallocate the memory.

Solution for Q5+Q6

CSC209H Worksheet: Stacks and Heaps

4. Trace the memory usage for the program below. We have set up the stack frame for you, and the location of the heap.

Value Address Label Section #include <stdio.h> 0x23c Heap 3 #include <stdlib.h> 6 0x228/* Build an array in dynamic memory to hold multiples of x from x to x*x. 0x22c Return a pointer to this array. 4 */ 0x230 int *multiples(int x) { int *a = malloc(sizeof(int) * x); 0x234 for (int i=0; i < x; i++) { 17_ a[i] = (i+1) * x;0x238return a; 0x23c 1 0x240 6 int main() { int *ptr; 0x24410 int size = 3; for (int j=size; j <= 5; i++) { stack frame 0x470 for multiples ptr = multiples(size); 0x4700x474for (int i=0; i<s22e; i++) { 0x478 printf("%d\t", ptr[i]); stack frame 0x47c printf("\n"); for main 0x480 0x4845120 return 0; 0x4883 O_{k} 23 c ptr 0x48c

- 5. Change the main function so that it calls multiples and prints the array in a loop with sizes of 3, 4, and 5. Besides the changes described, do not make any other changes or additions to the code.
- 6. Trace the memory usage of your changed program. Explain the problem to your partner and then fix it by adding calls to deallocate the memory.

 Free (p+r),