ghp\_mEHXX5RIWspSLsoTqigAZ7xJvOEcRU2Eg1Gu

int \_putchar(char);

void print\_alphabet(void);

void print\_alphabet\_x10(void);

int \_islower(int c);

int \_isalpha(int c);

int print\_sign(int n);

int \_abs(int);

int print\_last\_digit(int);

void jack\_bauer(void);

void times\_table(void);

int add(int, int);

void print\_to\_98(int n);

void print\_times\_table(int n);

#endif

#include "main.h"

/\*\*

\*main - print putchar

\*Return: always 0

\*/

int main(void)

{

\_putchar('\_');

\_putchar('p');

\_putchar('u');

\_putchar('t');

\_putchar('c');

\_putchar('h');

\_putchar('a');

\_putchar('r');

\_putchar('\n');

return (0);

2.

#include "main.h"

/\*\*

\*print\_alphabet - function to print abc

\*

\*Return: 0

\*/

void print\_alphabet(void)

{

char c;

for (c = 'a'; c <= 'z'; c++)

}^C

root@a93756abbbca:/alx-low\_level\_programming/0x02-functions\_nested\_loops# cat > 1-alphabeth.c

#include "main.h"

/\*\*

\*print\_alphabet - function to print abc

\*

\*Return: 0

\*/

void print\_alphabet(void)

{

char c;

for (c = 'a'; c <= 'z'; c++)

git clone https://github.com/AnitaNonye/alx-pre\_course.git

ghp\_ZYh4yjWlmKKHjM4BLgW8VPtj8B8wPP2hqPTZ

git clone https://ghp\_ZYh4yjWlmKKHjM4BLgW8VPtj8B8wPP2hqPTZ@github.com/Anita5810/alx-low\_level\_programming.git

#include "main.h"

/\*\*

\* print\_alphabet - Entry point

\* Return:0

\*/

void print\_alphabet(void)

{

char c = 'a';

while (c <= 'z')

{

\_putchar(c);

c++;

}

\_putchar('\n');

}

3.

#include "main.h"

/\*\*

\* \_islower - check for lower case letter

\* @c: character to check the case

\* Return:0 or 1

\*/

int \_islower(int c)

{

return (c >= 97 && c <= 122);

}

4

#include "main.h"

/\*\*

\* \_isalpha - check for lower case letter

\* @c : character to check

\* Return:0 or 1

\*/

int \_isalpha(int c)

{

return ((c >= 97 && c <= 122) || (c >= 65 && c <= 90));

}

5.sign

#include "holberton.h"

/\*\*

\* print\_sign - check numers

\* @n : number to check

\* Return:0 or 1

\*/

int print\_sign(int n)

{

int test;

if (n > 0)

{

test = 1;

\_putchar('+');

}

else if (n == 0)

{

test = 0;

/ \_putchar('0');

}

else

{

test = -1;

\_putchar('-');

}

return (test);

}

6.abs.c

#include "main.h"

/\*\*

\* \_abs - return abs value

\* @n : number to check

\* Return:0 or 1

\*/

int \_abs(int n)

{

return (n >= 0 ? n : n \* -1);

}

* 7. -print\_last\_digit.c
* #include "main.h"
* /\*\*
* \* print\_last\_digit - return last digit
* \* @n: number to check
* \* Return:0 or 1
* \*/
* int print\_last\_digit(int n)
* {
* int nv;
* if (n < 0)
* nv = -1 \* (n % 10);
* else
* nv = n % 10;
* \_putchar(nv + '0');
* return (nv);
* }
* 8. -24\_hours.c

 #include "main.h"

/\*\*

\* jack\_bauer - prints every minute of the day of Jack Bauer

\* Return:void

\*/

void jack\_bauer(void)

{

int h1;

int h2;

int m1;

int m2;

int a = 9;

h2 = 0;

while (h2 <= 2)

{

if (h2 == 2)

{

/\*Restrain to 23h, not 29\*/

a = 3;

}

h1 = 0;

while (h1 <= a)

{

m2 = 0;

while (m2 <= 5)

{

m1 = 0;

while (m1 <= 9)

{

\_putchar('0' + h2);

\_putchar('0' + h1);

\_putchar(':');

\_putchar('0' + m2);

\_putchar('0' + m1);

\_putchar('\n');

m1++;

}

m2++;

}

h1++;

}

h2++;

}

}

* }
* 9. times\_table.c

#include "main.h"

/\*\*

\* main - check the code for alx School students.

\*

\* Return: Always 0.

\*/

int main(void)

{

times\_table();

return (0);

}

9.

#include "main.h"

/\*\*

\* times\_table - prints times table

\* Return:void

\*/

void times\_table(void)

{

int a = 0;

int b;

int rep;

while (a <= 9)

{

b = 0;

while (b <= 9)

{

rep = a \* b;

if (b == 0)

{

\_putchar('0' + rep);

}

else if (rep < 10)

{

\_putchar(' ');

\_putchar('0' + rep);

}

else

{

\_putchar('0' + rep / 10);

\_putchar('0' + rep % 10);

}

if (b < 9)

{

\_putchar(',');

\_putchar(' ');

}

b++;

}

\_putchar('\n');

a++;

}

}

10. add.c

#include "main.h"

/\*\*

\* add - returns the sum of its parameters

\* @a: int type number

\* @b: int type number

\* Return:0

\*/

int add(int a, int b)

{

return (a + b);

}

11.print to 98

#include "main.h"

#include <stdio.h>

/\*\*

\* print\_to\_98 - print to 98

\* @n : number to start from

\* Return:0 or 1

\*/

void print\_to\_98(int n)

{

while (n < 98)

{

printf("%i, ", n);

n++;

}

while (n > 98)

{

printf("%i, ", n);

n--;

}

printf("98");

putchar('\n');

}

12 1—0

#include "main.h"

/\*\*

\* print\_times\_table - prints times table

\* @n : times table to use

\* Return:void

\*/

void print\_times\_table(int n)

{

int a = 0, rep, b;

if (n < 0 || n > 15)

return;

while (a <= n)

{

for (b = 0; b <= n; b++)

{

rep = a \* b;

if (b == 0)

\_putchar('0' + rep);

else if (rep < 10)

{

\_putchar(' ');

\_putchar(' ');

\_putchar('0' + rep);

}

else if (rep < 100)

{

\_putchar(' ');

\_putchar('0' + rep / 10);

\_putchar('0' + rep % 10);

}

else

{

\_putchar('0' + rep / 100);

\_putchar('0' + (rep - 100) / 10);

\_putchar('0' + rep % 10);

}

if (b < n)

{

\_putchar(',');

\_putchar(' ');

}

}

\_putchar('\n');

a++;

}

}

13

#include <stdio.h>

#include <stdlib.h>

/\*\*

\* main - main block

\* Description: computes and prints the sum of all the multiples of 3 or

\* 5 below 1024 (excluded), followed by a new line

\* Return: 0

\*/

int main(void)

{

int c = 0;

int sum = 0;

while (c < 1024)

{

if (c % 3 == 0 || c % 5 == 0)

{

sum += c;

}

c++;

}

printf("%i\n", sum);

return (0);

}

14 102 fibonacci

#include <stdio.h>

/\*\*

\* main - main block

\* Description: computes and prints the sum of all the multiples of 3 or

\* 5 below 1024 (excluded), followed by a new line

\* Return: 0

\*/

int main(void)

{

int i = 0;

long int a = 0, b = 1, next;

while (i < 50)

{

next = a + b;

a = b;

b = next;

printf("%lu", next);

if (i < 49)

{

printf(", ");

}

i++;

}

putchar('\n');

return (0);

}

15.103 fibonacci

#include <stdio.h>

/\*\*

\* main - main block

\* Description: computes and prints even number < 4,000,000

\* 5 below 1024 (excluded), followed by a new line

\* Return: 0

\*/

int main(void)

{

int a = 0, b = 1, next = 0;

int sum = 0;

while (next < 4000000)

{

next = a + b;

a = b;

b = next;

if (next % 2 == 0)

sum += next;

}

printf("%i\n", sum);

return (0);

}

16.104 fibonacci

#include <stdio.h>

/\*\*

\* main - main block

\* Description: computes and prints the sum of all the multiples of 3 or

\* 5 below 1024 (excluded), followed by a new line

\* Return: 0

\*/

int main(void)

{

int i = 0;

unsigned long int a = 0, b = 1, next = 0;

while (i < 98)

{

next = a + b;

a = b;

b = next;

printf("%lu", next);

if (i < 97)

printf(", ");

i++;

}

putchar('\n');

return (0);

}

#include "main.h"

/\*\*

\* print\_alphabet\_x10 - prints alphabet 10 times

\* Return:void

\*/

void print\_alphabet\_x10(void)

{

char c;

int i = 0;

while (i < 10)

{

c = 'a';

while (c <= 'z')

{

\_putchar(c);

c++;

}

\_putchar('\n');

i++;

}

}

1.debugging

#include "main.h"

/\*\*

\* main - Test function for positive or negative

\* Return: 0

\*/

int main(void)

{

int i;

i = 0;

positive\_or\_negative(i);

return (0);

}

2.main.c

#include <stdio.h>

/\*\*

\* main - causes an infinite loop

\* Return: 0

\*/

int main(void)

{

int i;

printf("Infinite loop incoming :(\n");

i = 0;

/\*

\*while (i < 10)

\*{

\* putchar(i);

\*}

\*/

printf("Infinite loop avoided! \\o/\n");

return (0);

}

3.2-largest

#include "main.h"

/\*\*

\* largest\_number - returns the largest of 3 numbers

\* @a: first integer

\* @b: second integer

\* @c: third integer

\* Return: largest number

\*/

int largest\_number(int a, int b, int c)

{

int largest;

if (a > b && a > c)

{

largest = a;

}

else if (b > c && b > a)

{

largest = b;

}

else if (c > b)

{

largest = c;

}

else

{

largest = b;

}

return (largest);

}