

```

#include <pthread.h>
#include <stdio.h>

void* happy (void* arg) {
    int* N = (int *)arg;
    for (int k = 0; k < *N; k++)
        printf ("Happy %d\n", k);

    return (void *) 0xF00D;
}

int main () {
    pthread_t happy_id;
    pthread_attr_t happy_attr;

    pthread_attr_init (&happy_attr);

    int threadArgument;
    threadArgument = 6;
    pthread_create (&happy_id, &happy_attr, happy, &threadArgument);

    for (int k = 0; k < 20; k++)
        printf ("Before %d\n", k);

    int joinReturnValue;
    pthread_join (happy_id, (void **) &joinReturnValue);
    printf ("Thread status is %x\n", joinReturnValue);

    printf ("End of program\n");

    return 0;
}

```

Execution

1. Implement the thread code to the left
2. When you run the code, what output do you get?
Does the output change over different iterations?
3. Change the number of times the happy worker method prints its message to 10. Make this change without modifying the happy method itself.
4. Change the return value for the 'happy' worker thread. Are there any limitations for what you can return?
5. Submit your modified code along with a word document under the in-Class Assignment