alloc1 実行結果

01_sample103-1.c

code

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

int main(void)
{
    char *str;
    int num, i;

    printf("num > ");
    scanf("%d", &num);

    str = (char *)malloc(sizeof(char)*(num+1));
    if(str==NULL) {
        printf("not allocated.\n");
        return 1;
    }

    for(i=0; i<num; i++) {
        *(str+i) = 'a';
    }
    *(str+i) = '\0';
    printf("str: %s\n", str);

    free(str);

    return 0;
}
```

result

Test 1 passed

入力:

```
10
```

出力:

```
num > str: aaaaaaaaaa
```

Test 2 passed

入力:

```
5
```

出力:

```
num > str: aaaaa
```

==========

- Passed: 2
- Failed: 0

02_sample103-2.c

code

```c
#include <stdio.h>

void show_range(int *ptr, int s, int e);

void show_range(int *ptr, int s, int e)
{
    int i;
    for(i=s; i<=e; i++) {
        printf("*ptr+%d: %d, ptr+%d: %p\n", i, *(ptr+i), i, ptr+i);
    }
}

int main(void)
{
    int test[5] = {80, 60, 55, 22, 75};
    printf("---show_range(test, 2, 4)---\n");
    show_range(test, 2, 4);
    printf("---show_range(test, 1, 3)---\n");
    show_range(test, 1, 3);
    return 0;
}
```

result

Test passed

入力:

```

```

出力:

```
---show_range(test, 2, 4)---
*ptr+2: 55, ptr+2: 0x7fffa399a138
```

```
*ptr+3: 22, ptr+3: 0x7fffa399a13c
*ptr+4: 75, ptr+4: 0x7fffa399a140
---show_range(test, 1, 3)---
*ptr+1: 60, ptr+1: 0x7fffa399a134
*ptr+2: 55, ptr+2: 0x7fffa399a138
*ptr+3: 22, ptr+3: 0x7fffa399a13c
```

==========

- Passed: 1
- Failed: 0

03_prog3-1.c

code

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

int main() {
    int *array;
    int num, i;
    printf("num > ");
    scanf("%d", &num);
    array = (int *)malloc(num * sizeof(int));
    if (array == NULL) {
        printf("Memory allocation failed\n");
        return 1;
    }

    for (i = 0; i < num; i++) {
        array[i] = 7;
    }
    for (i = 0; i < num; i++) {
        printf("%d ", array[i]);
    }
    printf("\n");

    free(array);
    return 0;
}
```

result

Test 1 passed

入力:

```
5
```

出力:

```
num > 7 7 7 7 7
```

Test 2 passed

入力:

```
10
```

出力:

```
num > 7 7 7 7 7 7 7 7 7 7
```

==========

- Passed: 2
- Failed: 0

04_prog3-2.c

code

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

char *make_string();

int main(void)
{
    char *mystr;
    mystr = make_string();
    printf("mystr: %s\n", mystr);
    free(mystr);
    return 0;
}


char *make_string()
{
    char *str;
    int num, i;

    printf("num > ");
    scanf("%d", &num);

    str = (char *)malloc(sizeof(char)*(num+1));
    if(str==NULL) {
        printf("not allocated.\n");
        return NULL;
    }

    for(i=0; i<num; i++) {
        *(str+i) = 'a';
    }
```

```
            *(str+i) = '\0';

            return str;
        }
```

result

Test 1 passed

入力:

```
8
```

出力:

```
num > mystr: aaaaaaaa
```

Test 2 passed

入力:

```
3
```

出力:

```
num > mystr: aaa
```

==========

- Passed: 2
- Failed: 0

05_prog3-3.c

code

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

        int *make_even(int num);

        int main(void)
        {
            int i;
            int *array;
            printf("---make_even(7)---\n");
            array = make_even(7);
            for(i=0; i<7; i++) {
            printf("%d ", *(array+i));
```

```
            }
            printf("\n");
            free(array);
            printf("---make_even(10)---\n");
            array = make_even(10);
            for(i=0; i<10; i++) {
            printf("%d ", *(array+i));
            }
            printf("\n");
            free(array);

            return 0;
        }

        int *make_even(int num)
        {
            int i;
            int *array;
            array = (int *)malloc(sizeof(int) * num);
            if(array == NULL) {
                printf("メモリの確保に失敗しました\n");
                exit(1);
            }
            for(i=0; i<num; i++) {
                *(array+i) = i * 2;
            }
            return array;
        }
```

result

Test passed

入力:

```
```

出力:

```
---make_even(7)---
0 2 4 6 8 10 12
---make_even(10)---
0 2 4 6 8 10 12 14 16 18
```

==========

- Passed: 1
- Failed: 0

06_prog3-4.c

code

```
```

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

char *fill_alpha(int num);

int main(void)
{
    char *mystr;

    printf("---fill_alpha(5)---\n");
    mystr = fill_alpha(5);
    printf("mystr: %s\n", mystr);
    free(mystr);
    printf("---fill_alpha(20)---\n");
    mystr = fill_alpha(20);
    printf("mystr: %s\n", mystr);
    free(mystr);
    return 0;
}

char *fill_alpha(int num)
{
    char *str;
    int i;

    str = (char *)malloc(sizeof(char) * (num + 1));
    if (str == NULL) {
        printf("メモリの確保に失敗しました\n");
        exit(1);
    }
    for (i = 0; i < num; i++) {
        str[i] = 'a' + i;
    }
    str[num] = '\0';

    return str;
}
```

result

Test passed

入力:

出力:

```
---fill_alpha(5)---
mystr: abcde
---fill_alpha(20)---
mystr: abcdefghijklmnopqrst
```

==========

- Passed: 1

- Failed: 0