

# Tutorial 1: C-Language

## Problem 1: Data Type

What are the ranges of the following data types?

int

long

long long

short

char

## Problem 2: Data Type Conversion

Consider the C code in Figure 1. Answer the following questions.

What are the meaning of statements `(b = (short) a;)` and `(c = (char) a;)`?

What are the outputs of the `printf` statements?

What are the outputs of the `printf` statements if `(a = -232590606)`?

```
#include <stdio.h>

int main (void)
{
    int a=-232644062;
    short b;
    char c;

    b = (short) a;
    c = (char) a;
    printf("integer = \"%d\"\n\n", a);
    printf("short = \"%d\"\n\n", b);
    printf("char = \"%d\"\n\n", c);

    return 0;
}
```

Figure 1: Program on Data Type Conversion

### Problem 3: More Data Type Conversion

Consider the C code in Figure 4. Answer the following questions.

What are the meaning of statements (a = (int) b;) and (c = (char) b;)?

What are the outputs of the printf statements?

What are the outputs of the printf statements if (b = -30848)?

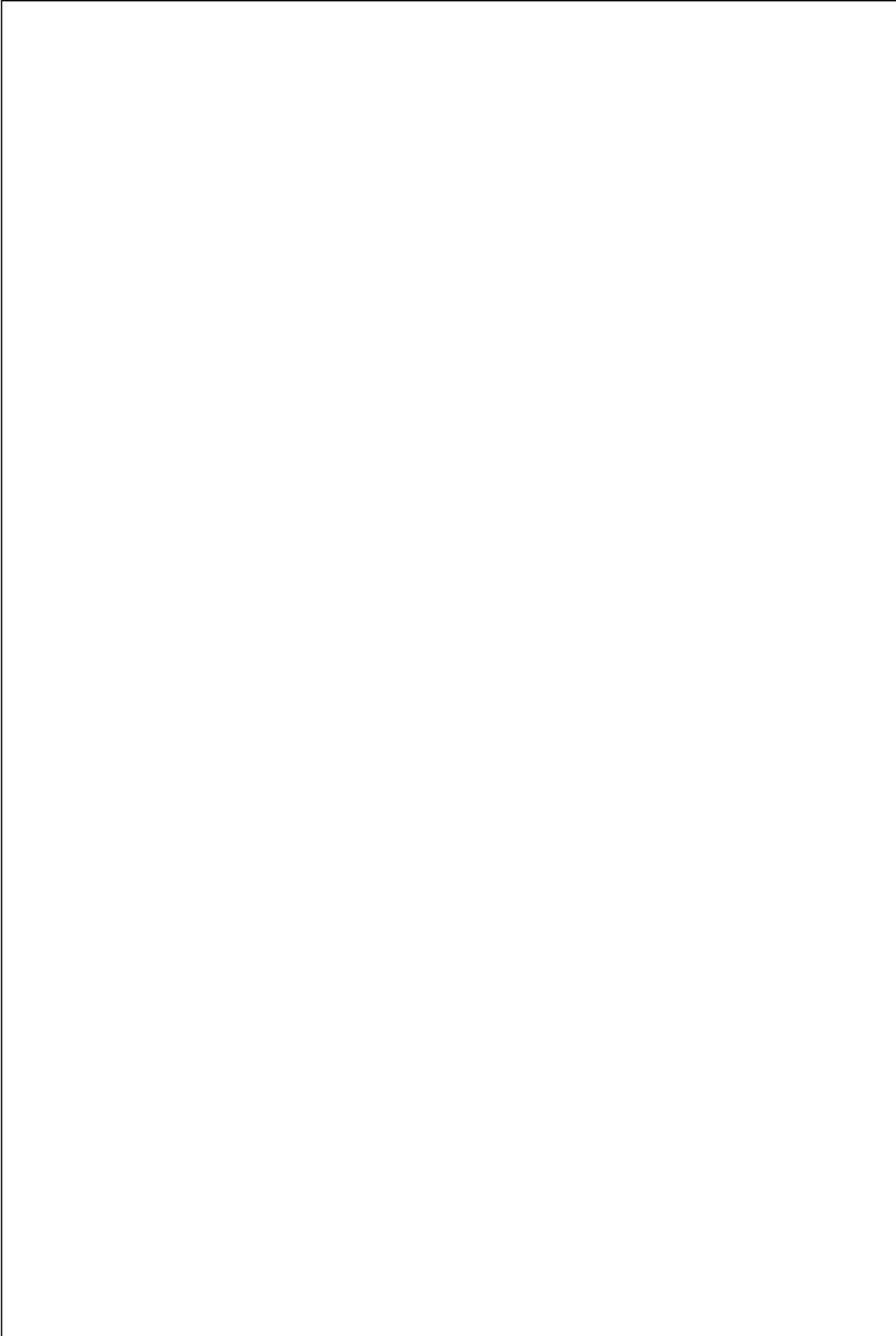
```
#include <stdio.h>

int main (void)
{
    short b= -32768;
    int a;
    char c;

    a = (int) b;
    c = (char) b;
    printf("short = \"%d\"\n\n", b);
    printf("integer = \"%d\"\n\n", a);
    printf("char = \"%d\"\n\n", c);

    return 0;
}
```

**Figure 4: Program on Data Type Conversion**



## Problem 5: Still Some More Data Type Conversion

Consider the C code in Figure 10. Answer the following questions.

What are the outputs of the `printf` statements?

What is the output of the last `printf` statement, if `(+ 1)` is removed from the statement `(d = (short *) &a + 1;)`?

```
#include <stdio.h>

int main (void)
{
    int a=-232644062;
    char *b, *c;
    short *d;

    b =(char *) &a;
    c =(char *) &a + 1;
    d =(short *) &a + 1;

    printf("char1 = \"%d\\\"\\n\\n", *b);
    printf("char2 = \"%d\\\"\\n\\n", *c);
    printf("short = \"%d\\\"\\n\\n", *d);

    return 0;
}
```

Figure 10: Program on Data Type Conversion with Pointers

