Исходный текст программы

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

typedef struct {
    unsigned char oct1;
    unsigned char oct2;
    unsigned char oct3;
    unsigned char oct4;
} IPv4Address;

void convertBtoC(IPv4Address* addrB);
void convertCtoB(IPv4Address* addrC);
void printIPv4Address(IPv4Address* addr);
void printIPv4AddressBinary(IPv4Address* addr);
void printBinary(int n);
void inputIpv4(IPv4Address* addr, char ipClass);

void clearConsole();
void pressEnterToContinue();
void clearStdin();

int main() {
    IPv4Address addr;

    clearConsole();

    inputIpv4(&addr, 'B');

    printf("Original Class B address: ");
    printIPv4Address(&addr);
    printIPv4AddressBinary(&addr);

    convertBtoC(&addr);
    printf("Converted to Class C: ");
    printIPv4Address(&addr);
    printIPv4AddressBinary(&addr);

    pressEnterToContinue();
```

```
    inputIpv4(&addr, 'C');

    printf("\nOriginal Class C address: ");
    printIPv4Address(&addr);
    printIPv4AddressBinary(&addr);

    convertCtoB(&addr);
    printf("Converted to Class B: ");
    printIPv4Address(&addr);
    printIPv4AddressBinary(&addr);

    pressEnterToContinue();

    return 0;
}

void inputIpv4(IPv4Address* addr, char ipClass) {
    int oct1 = 0, oct2 = 0, oct3 = 0, oct4 = 0;
    int condition;

    do {
        printf("Enter class %c ipv4 address: ", ipClass);
        scanf("%d.%d.%d.%d", &oct1, &oct2, &oct3, &oct4);
        clearStdin();
        if (ipClass == 'B') {
            condition = (128 <= oct1 && oct1 < 192) && (0 <= oct2 && oct2 < 256) && (0 <= oct3 && oct3 <
256) && (0 <= oct4 && oct4 < 256);
        } else {
            condition = (192 <= oct1 && oct1 < 224) && (0 <= oct2 && oct2 < 256) && (0 <= oct3 && oct3 <
256) && (0 <= oct4 && oct4 < 256);
        }
        if (condition) {
            addr->oct1 = (unsigned char)oct1;
            addr->oct2 = (unsigned char)oct2;
            addr->oct3 = (unsigned char)oct3;
            addr->oct4 = (unsigned char)oct4;
        } else {
            printf("Invalid class %c ipv4 address!\n", ipClass);
        }
    } while (!condition);
}
```

```
void clearConsole() {
    #if defined(_WIN32) || defined(_WIN64)
        system("cls");
    #else
        system("clear");
    #endif
}

void convertBtoC(IPv4Address* addrB) {
    /* 0x1F = 0001 1111 i.e. 31  ; we need to make 1st three bits = 0 and save other bits */
    /* 0xC0 = 1100 0000 i.e. 192 ; we need to make 1st two bits = 1 and save other bits */
    addrB->oct1 = (addrB->oct1 & 0x1F) | 0xC0;
}

void convertCtoB(IPv4Address* addrC) {
    /* 0x3F = 0011 1111 i.e. 63  ; we need to make 1st two bits = 0 and save other bits */
    /* 0x80 = 1000 0000 i.e. 128 ; we need to make 1st bit = 1 and save other bits */
    addrC->oct1 = (addrC->oct1 & 0x3F) | 0x80;
}

void printIPv4Address(IPv4Address* addr) {
    printf("%d.%d.%d.%d\n", addr->oct1, addr->oct2, addr->oct3, addr->oct4);
}

void clearStdin() {
    int c;
    while ((c = getchar()) != '\n' && c != EOF) { }
}

void pressEnterToContinue() {
    printf("\nPress ENTER to continue ");
    clearStdin();
    clearConsole();
}

void printIPv4AddressBinary(IPv4Address* addr) {
    printf("Address in binary: ");

    printBinary(addr->oct1);
```

```
    printf(".");

    printBinary(addr->oct2);
    printf(".");

    printBinary(addr->oct3);
    printf(".");

    printBinary(addr->oct4);
    printf("\n");
}

void printBinary(int n) {
    int i;
    for (i = 7; i >= 0; i--) {
        printf("%d", (n >> i) & 1);
    }
}
```

Контрольные примеры

Пример 1:

```
Enter class B ipv4 address: 145.120.8.0

Original Class B address: 145.120.8.0

Address in binary: 10010001.01111000.00001000.00000000

Converted to Class C: 209.120.8.0

Address in binary: 11010001.01111000.00001000.00000000


Enter class C ipv4 address: 220.168.1.1

Original Class C address: 220.168.1.1

Address in binary: 11011100.10101000.00000001.00000001

Converted to Class B: 156.168.1.1

Address in binary: 10011100.10101000.00000001.00000001
```

Пример 2:

```
Enter class B ipv4 address: 0.0.0.0.0

Invalid class B ipv4 address!

Enter class B ipv4 address: 128.-2.2500.0

Invalid class B ipv4 address!

Enter class B ipv4 address: 192.0.0.0

Invalid class B ipv4 address!

Enter class B ipv4 address: 128.168.1.1

Original Class B address: 128.168.1.1

Address in binary: 10000000.10101000.00000001.00000001

Converted to Class C: 192.168.1.1

Address in binary: 11000000.10101000.00000001.00000001
```

```
Enter class C ipv4 address: 128.0.0.0

Invalid class C ipv4 address!

Enter class C ipv4 address: 192.168.1.1


Original Class C address: 192.168.1.1

Address in binary: 11000000.10101000.00000001.00000001

Converted to Class B: 128.168.1.1

Address in binary: 10000000.10101000.00000001.00000001
```

Примеры выполнения программы

```
Enter class B ipv4 address: 145.120.8.0
Original Class B address: 145.120.8.0
Address in binary: 10010001.01111000.00001000.00000000
Converted to Class C: 209.120.8.0
Address in binary: 11010001.01111000.00001000.00000000
```

```
Enter class C ipv4 address: 220.168.1.1

Original Class C address: 220.168.1.1
Address in binary: 11011100.10101000.00000001.00000001
Converted to Class B: 156.168.1.1
Address in binary: 10011100.10101000.00000001.00000001
```

```
Enter class B ipv4 address: 0.0.0.0.0
Invalid class B ipv4 address!
Enter class B ipv4 address: 128.-2.2500.0
Invalid class B ipv4 address!
Enter class B ipv4 address: 192.0.0.0
Invalid class B ipv4 address!
Enter class B ipv4 address: 128.168.1.1
Original Class B address: 128.168.1.1
Address in binary: 10000000.10101000.00000001.00000001
Converted to Class C: 192.168.1.1
Address in binary: 11000000.10101000.00000001.00000001
```

```
Enter class C ipv4 address: 128.0.0.0
Invalid class C ipv4 address!
Enter class C ipv4 address: 192.168.1.1

Original Class C address: 192.168.1.1
Address in binary: 11000000.10101000.00000001.00000001
Converted to Class B: 128.168.1.1
Address in binary: 10000000.10101000.00000001.00000001
```

Выводы.

В результате выполнения работы изучены методы работы с битовыми полями структур; получены практические навыки при программировании на языке С.