

```

parseInput_1(){
    parser = new ArgumentParser()
    parser.add_argument('--epochs', type=int, default=500)
    parser.add_argument('--max_time', type=int, default=10)
    parser.add_argument('--test_steps', type=int, default=10)
    parser.add_argument('--batch_size', type=int, default=20)
    parser.add_argument('--data_dir', type=str, default='data/s2s_mitbih_aami')
    parser.add_argument('--bidirectional', type=str2bool, default=str2bool('False'))
    parser.add_argument('--num_units', type=int, default=128)
    parser.add_argument('--n_oversampling', type=int, default=10000)
    parser.add_argument('--checkpoint_dir', type=str, default='checkpoints-seq2seq')
    parser.add_argument('--ckpt_name', type=str, default='seq2seq_mitbih.ckpt')
    parser.add_argument('--classes', nargs='+', type=chr, default=['F','N', 'S','V'])
    args = parser.parse_args()
}

run_program_1(){
    read_data_1()
    splitTrainTest_1()
    dataAugmentation()
    buildNetwork()
    train()
    test()
}

main(){
    parseInput_1()
    run_program_1()
}

```

```

parseInput_2(){
    parser = new ArgumentParser()
    parser.add_argument('--epochs', type=int, default=500)
    parser.add_argument('--max_time', type=int, default=10)
    parser.add_argument('--test_steps', type=int, default=10)
    parser.add_argument('--batch_size', type=int, default=20)
    parser.add_argument('--data_dir', type=str, default='data/s2s_mitbih_DS1DS2')
    parser.add_argument('--bidirectional', type=str2bool, default=str2bool('False'))
    parser.add_argument('--num_units', type=int, default=128)
    parser.add_argument('--n_oversampling', type=int, default=6000)
    parser.add_argument('--checkpoint_dir', type=str, default='checkpoints-seq2seq_DS1DS2')
    parser.add_argument('--ckpt_name', type=str, default='seq2seq_mitbih_DS1DS2.ckpt')
    parser.add_argument('--classes', nargs='+', type=chr, default=['F','N', 'S','V'])
    args = parser.parse_args()
}

run_program_2(){
    read_data_2()
    splitTrainTest_2()
    dataAugmentation()
    buildNetwork()
    train()
    test()
}

main(){
    parseInput_2()
    run_program_2()
}

```

