NATIONAL UNIVERSITY OF HO CHI MINH CITY

PROJECT REPORT

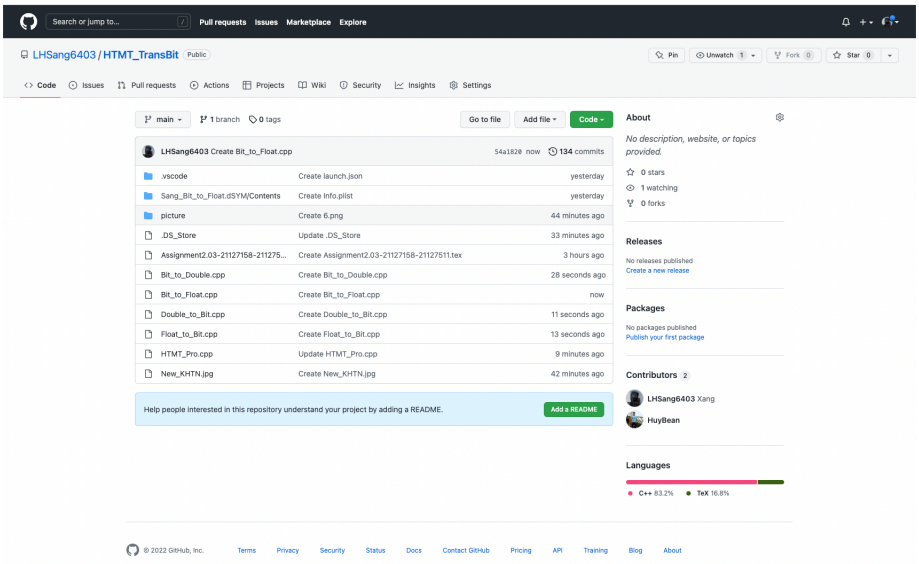
21CLC02 - Lê Hoàng Sang - 21127158

21CLC02 - Nguyễn Quốc Huy - 21127511

HỆ THỐNG MÁY TÍNH

1 PHẦN 1: LỜI NÓI ĐẦU

- Đây là bài báo cáo về bài tập chuyển đổi cơ số Hệ Thống Máy Tính.
- Yêu cầu chương trình chạy trên console.
- Các thư viện cần thiết đã được nêu rõ trong chương trình.
- Sinh viên thực hiện: Lê Hoàng Sang - 21127158
- Sinh viên thực hiện: Nguyễn Quốc Huy - 21127511
- Mã nguồn của 2 bạn được đồng bộ qua Github



- Nguyễn Quốc Huy: Giao diện, báo cáo, các hàm chuyển đổi.
- Lê Hoàng Sang: Báo cáo, các hàm chuyển đổi, Github.

Contents

2 PHẦN 2: HƯỚNG DẪN

Hướng dẫn các thao tác trên chương trình:

- Đầu tiên trên chương trình sẽ hiện Console mở đầu, người dùng cần **sử dụng phím mũi tên lên xuống hoặc phím 'w' hay 's'** để di chuyển đến lựa chọn của mình.
  Sau đó nhấn dấu cách (spacebar) để xác nhận.

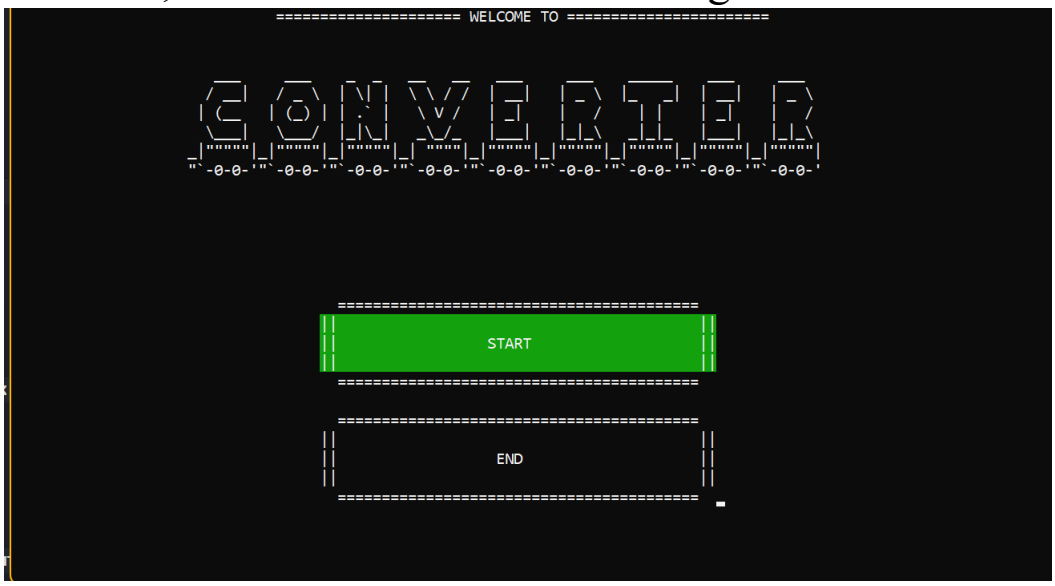
- Tiếp theo có các lựa chọn hiện ra như: Unsigned Char, Signed Short, Unsigned Int, Float,... đồng nghĩa với việc chuyển đổi dựa trên các kiểu dữ liệu tương ứng.
  Người dùng cần trượt thanh trên màn hình console để thấy hết lựa chọn và thực hiện thao tác sử dụng phím mũi tên lên xuống hoặc phím w, s để di chuyển và nhấn spacebar để chọn.

- Sau mỗi kiểu dữ liệu đã chọn, trên màn hình sẽ tiếp tục hiện ra lựa chọn:
  Binary to Decimal: Nhập dãy bit và cho chương trình chuyển về hệ thập phân.
  Decimal to Binary: Nhập số ở hệ thập phân để chương trình chuyển về dạng bit.

- Sau đó, chỉ cần thực hiện phép nhập số cho chương trình theo yêu cầu đã chọn.

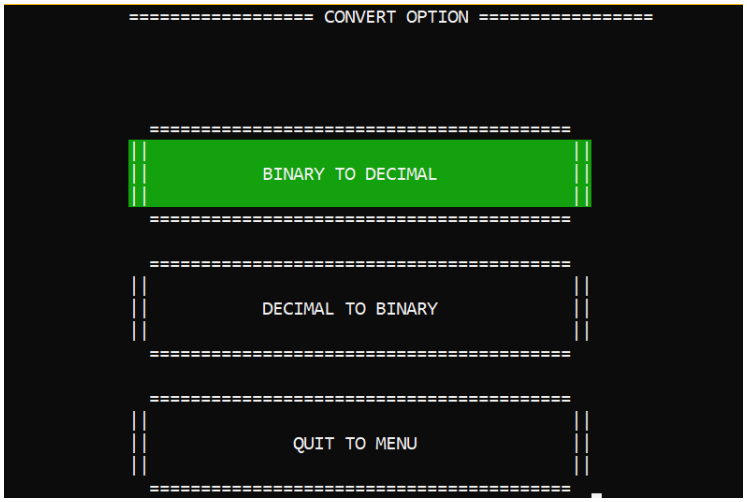
3 PHẦN 3: CÁC HÌNH MINH HỌA

Sau đây là chi tiết các bước có được qua màn hình console. Đầu tiên, màn hình mở đầu của chương trình.

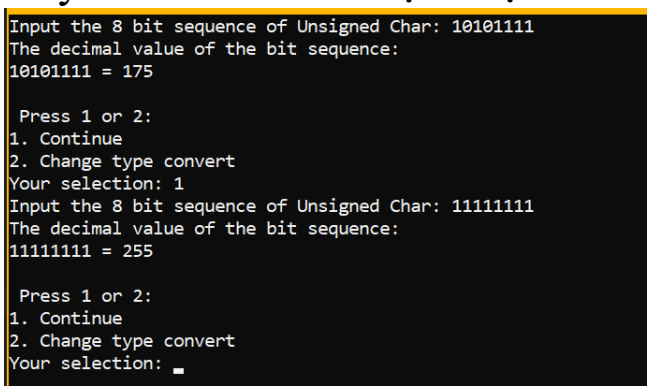


Kế đến là màn hình menu tùy chọn, người dùng cần trượt thanh trên console để thấy hết tùy chọn và thực hiện thao tác di chuyển phím mũi tên hoặc phím w, s và nhấn spacebar để chọn.

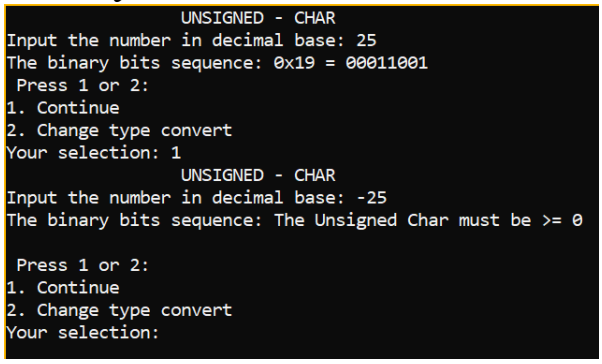
Tiếp theo là màn hình chọn chức năng chuyển đổi.



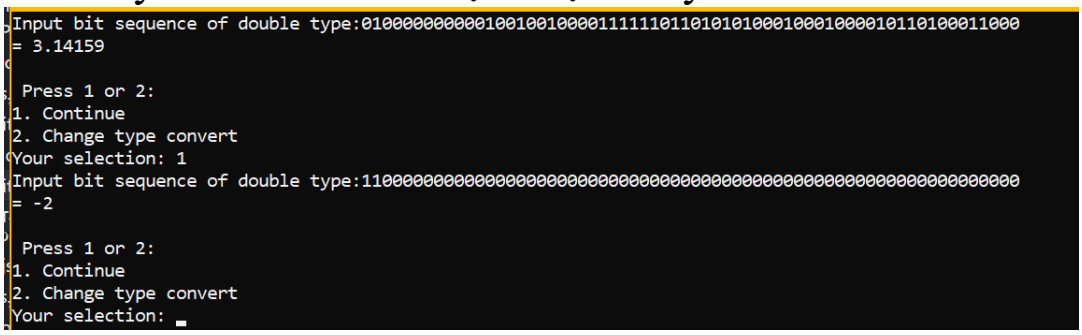
Đây là hình ảnh cho lựa chọn **BINARY TO DECIMAL**.



Và đây là hình ảnh cho lựa chọn **DECIMAL TO BINARY**.



Còn đây là hình ảnh cho lựa chọn chuyển đổi **SỐ CHẤM ĐỘNG**.



4 PHẦN 4: MÔ TẢ CÁC HÀM

4.1 HÀM ĐÁNH DẤU

- void $gotoxy$(short x, short y): Dùng để đưa các ký tự mong muốn đến tọa độ x y tương ứng trên console.

4.2 HÀM CHUYỂN ĐỔI ĐƠN GIẢN

- int $Convert$(char c): dùng để chuyển đổi các ký tự c nhập vào thành số nguyên trong trường hợp base > 10.
  Vd : $'A' = 10, 'B' = 11$....

- double $toDeci$(char *str, int base): dùng để chuyển chuỗi nhập vào của một base cho trước (2 <= base <= 32) thành hệ cơ số 10.
  Chi tiết thuật toán:
  Sử dụng hàm Convert bên trên để đổi từng ký tự thành số .
  Sau đó, tính tổng các phần tử sao cho mỗi phần tử sẽ nhân với số mũ mà sau mỗi lần cộng một phần tử vào, số mũ sẽ tăng lên base lần (số mũ lúc đầu = 1 do $base^0 = 1$).
  Quá trình tính sẽ được thực hiện từ phải sang trái.
  Vd :AAB_{16} = 11x1 + 10x16 + 10x16x16 = 2731_d

4.3 HÀM LÀM ĐẦY

- void $fillbit$(long long x, int bit, int Signed): Dùng để bổ sung các ký tự còn thiếu trên dãy tương ứng vớt bit.
  Vd: $x = 111100, bit = 8, Signed = 1 \rightarrow 00111100$
  Vd: $x = 101011, bit = 16, Singed = -1 \rightarrow 1111111111101011$

- void $fillbitstring$(string str, int bit, int Signed): Tương tự như chức năng hàm trên nhưng áp dụng cho chuỗi string.

4.4 HÀM CHUYỂN ĐỔI TỪ CHUỖI

- string $findTwoscomplement$(string str): Dùng để chuyển chuỗi của một số âm(nếu có) sang bù 2.
  Chi tiết thuật toán:
  Tìm vị trí mà tại đó số 1 đầu tiên tính từ bên phải xuất hiện và đánh dấu.
  Thêm số 1 ở đầu chuỗi (bit cực trái)
  Sau đó, tính từ vị trí số 1 đã đánh dấu, đảo ngược các bit phía sau: 1 = 0 và 0 = 1 cho đến khi gặp bit cực trái.

- string $ConvertFromDecimal$(long long from, int to): hàm này sử dụng đệ quy để đổi một số cho trước (from) từ base 10 sang base tùy chọn (to) và 2 <= base <= 32.
  Chi tiết thuật toán:
  Tạo một biến tạm để lưu kết quả (res) sau mỗi vòng.
  Lấy số của hệ 10 chia lấy dư cho base và gán vào một biến (digit).
  Sau đó, đổi các số dư digit sang ký tự, nếu <10 thì đổi thành ký tự từ 1 đến 9 nếu >= 10 thì đổi thành A, B, C,…
  Tiếp theo gán ký tự vào biến kết quả để thành một chuỗi hoàn chỉnh.
  Sau mỗi vòng, số (from) sẽ giảm dần theo công thức:
  from = from / base cho đến khi from = 0 (đã xử lý xong) thì dừng hàm.

- string toBase16(unsigned num): chuyển một số nhập vào chuyển thành hệ 16 trong chuỗi.

4.5 HÀM CHUYỂN TỪ DÃY BIT SANG SỐ

- void $BitToDecimal$(int bit, int Signed): Yêu cầu người dùng nhập một số và chương trình sẽ chuyển về dạng nhị phân. Biến signed để xác định xem liệu kiểu dữ liệu đó là unsigned(không dấu) hay signed(có dấu).

4.6 HÀM CHUYỂN TỪ CHUỖI BIT SANG DẠNG SỐ CHẤM ĐỘNG

- void $format\_M$(char $*bin\_arr$, int &$size$): Dùng để rút gọn dãy bit, cụ thể ở đây là phần M(mantissa), nếu phía sau thừa số 0 thì có thể rút ngắn bớt.
- void $swap\_2Char$(char &a, char &b): Đảo vị trí 2 ký tự, đây là hàm bổ sung sẽ được áp dụng vào những hàm sau.
- char $*add0\_Fisrt$(char *M, int &M_size, int p_idx, int E_val, int &$count\_multi$): trả về chuối có thêm các số 0 ở đầu chuỗi trong trường hợp E âm.
- void $multiply\_2Bin\_positive$(char *M, int M_size, int E_val): nhân 2 chuỗi Bits trong trường hợp E dương.
- char $*multiply\_2Bin\_negative$(char *M, int &M_size, int E_val): nhân 2 chuỗi Bits trong trường hợp E âm.
- float $BinToNum\_f$(char *M, $intM\_size$): chuyển từ Binary về dạng số, trường hợp Float.
- float $BinTo\_FloatPoint$(char *arr): xuất ra số Float hoàn chỉnh.
- double $BinToNum\_d$(char *M, int M_size): chuyển từ Binary về dạng số, trường hợp Double.
- double $BinTo\_DoublePoint$(char *arr): xuất ra số Double hoàn chỉnh.

4.7 HÀM CHUYỂN TỪ DẠNG SỐ CHẤM ĐỘNG SANG CHUỖI BIT

- int $countBit\_NeedMore\_f$(float input): trả về số Bits cần thêm.
- int $countBit\_NeedMore\_d$(float input): trả về số Bits cần thêm.
- long long $multiplyy$(float input): nhân số Bits cần thêm.
- char $*add\_pointBin$(long long input): thêm dấu chấm vào chuỗi Bits.
- int $count\_Exponent\_afterPoint$(char *res): đếm các Bits để tính E.
- int $count\_E\_value$(int exponent, int needMorebit): tính E.
- void $export\_bin\_f$(float input, char $*E\_arr$, char *man): xuất ra chuỗi Bits.
- void $export\_bin\_f$(float input, char $*E\_arr$, char *man): xuất ra chuỗi Bits.

4.8 HÀM ĐỔI TỪ DECIMAL SANG BIT

- void $Unsigned\_Char$(long long x): Dùng để kiểm tra tính hợp lệ của số nhập vào thỏa điều kiện Unsigned_Char hay không, sau đó chương trình sẽ chuyển sang dạng nhị phân.

- void $Signed\_Char$(long long x): Dùng để kiểm tra tính hợp lệ của số nhập vào thỏa điều kiện Unsigned_Char hay không, sau đó chương trình sẽ chuyển sang dạng nhị phân.

- void $Unsigned\_Short$(long long x): Dùng để kiểm tra tính hợp lệ của số nhập vào thỏa điều kiện Unsigned_Short hay không, sau đó chương trình sẽ chuyển sang dạng nhị phân.

- void $Signed\_Short$(long long x): Dùng để kiểm tra tính hợp lệ của số nhập vào thỏa điều kiện Signed_Short hay không, sau đó chương trình sẽ chuyển sang dạng nhị phân.

- void $Unsigned\_Int$(long long x): Dùng để kiểm tra tính hợp lệ của số nhập vào thỏa điều kiện Unsigned_Int hay không, sau đó chương trình sẽ chuyển sang dạng nhị phân.

- void $Signed\_Int$(long long x): Dùng để kiểm tra tính hợp lệ của số nhập vào thỏa điều kiện Signed_Int hay không, sau đó chương trình sẽ chuyển sang dạng nhị phân.

4.9 HÀM MENU

- void $INTRO$(int &key): Dùng để tạo menu console mở đầu.

- void $INTRO\_NEXT$(int &key): Dùng để tạo menu console cho các tùy chọn.

- $OPTION$(int &$BinToDec$): Tạo menu hỏi rằng liệu người chơi nhập $BinarytoDecimal$ hay $DecimaltoBinary$.

- void $ConvertAgain$(int &$Again$): Hỏi rằng người chơi có muốn tiếp tục convert hay đổi sang tùy chọn khác.

5 PHẦN 5: TÀI LIỆU THAM KHẢO

- `https://www.binaryconvert.com/convert_double.html`

- https://en.wikipedia.org/wiki/Double-precision_floating-format

- https://en.wikipedia.org/wiki/Single-precision_floating-format

- https://www.google.com/search?q=float+to+bit&sxsrf=ALiCzsbDJ8EnsSrW7UlVZsBpkWm3qBS5Rw%3A1654418958742&ei=Dm6cYvn4LP-MseMPyMyAkAw&ved=0ahUKEwj5t-Hy9pX4AhV_RmwGHUgmAMIQ4dUDCA4&uact=5&oq=float+to+bit&gs_lcp=Cgdnd3Mtd2l6EAMyBggAEB4QBzIFCAAQywEyBQgAEMsBMgUIABDLATI
sclient=gws-wiz

HẾT