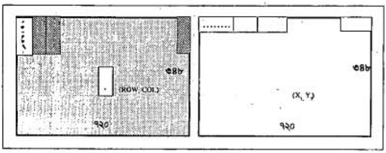
কিভাবে গ্রাফিক্স প্রিন্ট করবেন

॥ জাকারিয়া স্বপন ॥

কমপিউটার জগৎ-এর সাথে যুক্ত হবার পর থেকে আমি অনেক চিঠি ও ব্যক্তিগত অনুরোধ পেয়েছি — কিভাবে ইমেজ বা ছবি ডট প্রিন্টারে প্রিন্ট করা যায়, তা জানানোর জন্যে। আমরা যারা কমপিউটারে কাজ করি তারা সবাই কমবেশী গ্রাফিক্স-এ কাজ করে থাকি। পর্দায় সুন্দর সুন্দর ডিজাইন করি। অনেকে আবার স্ক্যানার দিয়ে নিজের ছবি টেনে নিয়ে কমপিউটারের পর্দায় তা নতুন করে ফুটিয়ে তুলি। গ্রাফিক্স-এর উপর যে কাজগুলোই আপনি করুন না কেন— নিশ্চই কখনও না কখনও আপনার মনে একটা ইচ্ছে জেগেছে— যদি এগুলো প্রিন্ট করা যেত। অনেকে হয়ত আবার কী বোর্ডের "স্ক্রিন প্রিন্ট" কী-টি চেপেও চেষ্টা করেছেন।

বাংলা ওয়ার্ড প্রসেসর আইবিএম কমপাটিবল মেশিনে অনেকেই দেখেছেন। এগুলোর চেয়েও সুন্দর ইমেজ/ফন্ট হয়তো অনেকেই ঘরে বসে তৈরী করতে পারবেন। এই "অ" "আ" "ক" "খ" -এর ইমেজগুলো যদি প্রিন্টারে পাঠাতে পারতেন তবে হয়তো আপনি নিজেও একটি ওয়ার্ড প্রসেসরের সমতুল্য কাজে হাত দিতে পারতেন।



এবার আসুন দেখি—একটা ইমেজকে বা পুরো স্ক্রীনটিকে প্রিন্টারে পাঠাতে গেলে আমাদের কি করতে হবে।

গ্রাফিক্স মোডে পর্দায় প্রতিটি বিন্দুকে বলে পিক্সেল। প্রতিটি পিক্সেলকে আমরা (x, y) — স্থানাংক দিয়ে প্রকাশ করে থাকি। প্রথমেই আপনাকে স্ক্রীনের রেজুলেশন জানতে হবে। সাধারণভাবে ব্যবহৃত স্ক্রীনগুলোর রেজুলেশন ৭২০ × ৩৮৪। আমি এটাকেই আমার লেখার আদর্শ ধরে নিচ্ছি। স্ক্রীনে যে বিন্দুটির অবস্থান (x_1, y_1) তাকে প্রিন্টারের যথার্থ স্থানাংকে পাঠাতে পারলেই আমাদের কাজ অনেকটা সহজ হয়ে গেলো। আসলে স্ক্রীনের প্রতিটি বিন্দুকে প্রিন্টারের নিজস্ব ফরম্যাট অনুযায়ী প্রিন্টার বাফারে পাঠাতে হবে। এবার জেনে নেয়া যাক — প্রিন্টার তার বাফারে ডাটা গুলো কিভাবে পাঠালে যথাবিহিত প্রিন্ট করে দেবে। চিত্রটি লক্ষ্য করুন —

পুরো স্ক্রীনে x-অক্ষ বরাবর ৭২০টি বিন্দু এবং y অক্ষ বরাবর ৩৮৪টি বিন্দু রয়েছে। প্রিন্টারে আমরা এক বাইট করে ডাটা পাঠাবো। এখন প্রশ্ন হলো — কোন আটটি বিট/বিন্দু/পিক্সেল নিয়ে গঠিত হবে এই বাইটটি। সাধারণ নিয়মে পরপর ৮টি বিন্দুকে একটা বাইট ধরার কথা (ডান পার্শের চিত্র)। এই নিয়মে ৭২০/৮ = ৯০ টি বাইট পাঠালে পুরো একটা লাইন টানা হয়ে যাবে।

কিন্তু প্রিন্টারের বেলায় সমস্যাটা এখানেই। প্রিন্টার সাধারণতঃ ৯ পিন বা ২৪ পিনের হয়ে থাকে। সে একই সাথে ৮টি ডট কাগজে ফেলতে পারে। তাই প্রিন্টারে বাইট পাঠাতে হবে উপর-থেকে নীচ পর্যন্ত-৮টি বিট দিয়ে গঠিত বাইট (বাপার্শের চিত্র)। অর্থাৎ ৭২০টি বাইট পাঠালে আপনি একই সাথে ৮টি লাইন ছাপা অবস্থায় পাবেন। একটি পুরো লাইন ছাপতে আপনাকে ৯০ বাইটের পরিবর্তে ৭২০ বাইট পাঠাতে হচ্ছে ; বিনিময়ে পাচ্ছেন অতিরিক্ত ৭টি লাইন।

স্ক্রীনের কোন বিন্দুকে প্রিন্টার বাফারে পাঠাতে কিছুটা হিসাবের ব্যাপার রয়েছে। প্রিন্টারের স্থানাংককে (x, y) -এর পরিবর্তে (row, col) স্থানাংক দ্বারা প্রকাশ করা সহজ এবং সহজ বোধ্য। গ্রাফিক্স মোডে আপনি তাই পাচ্ছেন ৭২০টি কলাম এবং ৩৪৮/৮ = ৪৪ টি রো (row)।

আমরা একটা two-dimensional array নেবো, যাকে আমরা বলবো p_buffer এবং এর মাত্রা হবে — unsigned char p_buffer [ROW_SIZE] [COL_SIZE];

এখানে ROW_SIZE ও COL_SIZE কে define করা হবে যথাক্রমে ৩৪৮/৮ এবং ৭২০ দ্বারা। p_buffer হবে unsigned char, যাকে আমরা ইতিপূর্বে বাইট বললাম। এজন্যে ৪৪×৭২০টি বাইট প্রয়োজন পড়বে।

একটা ব্যাপার লক্ষ্য করুন — x দ্বারা যা বুঝানো হয়, row দ্বারা কিন্তু তা বুঝায় না ; তা বুঝায় col (column এর সংক্ষিপ্ত রূপ) দ্বারা একই ভাবে y এর অনুরূপ অর্থ col নয়, row।

প্রথমে আমরা একটা procedure লিখবো — যা যে কোন একটি বিন্দু $P(x_1, y_1)$ কে p_buffer-এর যথার্থ row ও col এর বাইটে উক্ত বিন্দুটিকে স্থাপন করে দেবে। পরে এই procedure টিকে আমরা পুরো স্ক্রীনের প্রতিটি বিন্দুর জন্যে কাজে লাগিয়ে p_buffer-কে পুরো করবো। পরবর্তীতে সম্পূর্ণ p_buffer কে লুপের মাধ্যমে একটি একটি বাইট করে প্রিন্টারে পাঠাবো। গ্রাফিক্স মোডে কখনই string পাঠাবো না, সব সময় char পাঠাবো তা না হলে কোন শূন্য বাইটকে string terminator হিসেবে কম্পাইলার ধরে নেবে, তাতে আপনি পুরো ইমেজ পাবেন না।

আমি এখানে $P(x_1, y_1)$ একটি বিন্দুকে (row, col) - স্থানাংকে পরিবর্তন করার এলগরিদম ব্যাখ্যা করার চেষ্টা করছি।

P বিন্দুর x - স্থানাংক x_1 হলে, এটি অবশ্যই x_1 তম কলামে (col) স্থান পাবে এতে কোন সন্দেহ নেই। এবারে y_1 - নিয়ে ভাবা যাক। প্রথমেই একটা উদাহরণ নেই। ধরুন y_1 - এর মান 5 ; এক্ষেত্রে এটি অবশ্যই প্রথম লাইনের (row) ৫ম বিটে স্থান পাবে। y_1 এর মান যদি হয় ৮, তবে তা প্রথম লাইনের ৮ম বিটে স্থান পাবে। y_1 এর মান ৯ হলে তা ২য় লাইনের ১ম বিটে স্থান পাবে। অর্থাৎ ব্যাপারটি দাড়াচ্ছে এরকম $row = 1 + y_1/9$ এবং $bit = 1 + y_1 \bmod 9$; এখানে row বুঝাচ্ছে লাইন নাম্বার এবং bit বুঝাচ্ছে — উক্ত লাইনের যত তম বিট। যেহেতু আমি টার্বো-সি-তে প্রোগ্রাম লিখে দেবো, তাই সি-এর কনভেনশনটা এখানে ব্যাখ্যা করছি। সি-তে ১ম লাইনকে বলা হয় row = 0, ২য় লাইন হলো row = 1 একইভাবে ৫ম বিট মানে bit = 4. সেক্ষেত্রে উপরের হিসেবটি একটু পরিবর্তিত হবে। $row = y_1/8$ এবং bit = y1% 8(সি-তে modকে % দিয়ে লিখা হয়)।

এখন আমরা row, col এবং row - এর কততম bit টিতে একটি বিন্দু রাখতে হবে, তা পেয়ে গেলাম। এখানে বিন্দু রাখা মানে উক্ত বিটটিতে 1 (ওয়ান) বসানো। কোন একটি বাইটের নির্দিষ্ট কোন বিট 1 করতে হলে নতুন একটি বাইটে ঐ বিটে 1 রেখে তার সাথে পূর্বের বাইটের লজিকাল 'অর' (Logical OR operation) অপারেশন করতে হবে। যেমন — mask নামের একটি char - এ \x80 (হেক্স ৮০, ডেসিমেল ১২৮) রেখে OR করলে ১ম বিটটি সেট হয়ে যাবে, (১২৮ = ১০০০ ০০০০ বাইনারী)।

এই কাজটি করার জন্যে এখন আমি একটি array নেবো, যা নীচে দেখানো হলো।

```
unsigned char mask [ ] = { 128, 64, 32, 16,
8, 4, 2, 1 } ;
```

নীচে আমি প্রসিডিউরটি সি-এ লিখে দিলাম।

```
Void p_putpixel ( int x, int y)
int row, col, bit ;
col = x ;
row = y/8 ; bit = y%8 ;
p_buffer [row] [col] =
p_buffer[row] [col] | mask [ bit ] ;
                    /* Logical OR */
}
```

প্রথমেই p_buffer কে লুপ চালিয়ে শূন্য দিয়ে initialize করে নিন। এবারে স্ক্রীনের যে সকল স্থানে 1 পাবেন তাদেরকে উপরোক্ত প্রসিডিউর চালিয়ে p_buffer এর যথার্থস্থানে পাঠিয়েদিন। তারপর লুপ দিয়ে পুরো p_buffer কে একে একে প্রিন্টারে পাঠানোর ব্যবস্থা করুন। এগুলো করতে আপনার আরো যা করতে হবে যেমন, প্রিন্টার ইনিশিয়ালাইজ করা, ভার্টিকাল স্পীপ ঠিক করা ইত্যাদি পরবর্তী সংখ্যায় দেয়া হবে।